

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено
Завідувач кафедри
О.В. Коваль

“ ” 2019р.

ДИПЛОМНА РОБОТА
першого рівня вищої освіти з напрямку підготовки
6.050101 “Комп’ютерні науки”

на тему : “Інформаційне середовище кафедри АПЕПС на базі Office 365”

Виконав (-ла): студент (-ка) 4 курсу, групи ТР-51

Купріянов Іван Степанович

Керівник доц., к.т.н. Крамар Ю.М.

Рецензент доц., к.т.н. Іванова Л.М.

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2019

**Національний Технічний Університет України
«Київський Політехнічний Інститут ім. Ігоря Сікорського»**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль

“ ____ ” _____ 2019р.

ЗАВДАННЯ

на дипломну роботу студенту

Купріянову Івану Степановичу

1. Тема роботи “Інформаційне середовище кафедри АПЕПС на базі Office 365”

керівник роботи доц., к.т.н. Крамар Юлія Михайлівна

затверджена наказом вищого навчального закладу від 22.05.2019 р. №1325-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи середовище розробки Microsoft Visual Studio Code, мова програмування TypeScript.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) провести огляд методів розробки інформаційних середовищ; провести огляд та обґрунтувати вибір засобів розробки програмного продукту; розробити програмне забезпечення відповідно до теми і провести опис програмної реалізації; розробити методику роботи користувача з програмою.

5. Перелік графічного матеріалу архітектура системи, графічне представлення інтерфейсу, приклади роботи програмного модулю.

6. Дата видачі завдання “15 жовтня 2018 р”.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	14.10.2018-23.12.2018	
2.	Розробка архітектури та загальної структури системи	2.02.2019-3.03.2019	
3.	Розробка структур окремих підсистем	4.03.2019-14.04.2019	
4.	Підготовка матеріалів	15.04.2019-18.04.2019	
5.	Програмна реалізація системи	18.04.2019-14.05.2019	
6.	Захист програмного продукту	15.05.2019	
7.	Оформлення пояснювальної записки	16.05.2019-3.06.2019	
8.	Передзахист	28.05.2019	
9.	Захист	17.06.2019-22.06.2019	

Студент _____ Купріянов І.С.

Керівник роботи _____ Крамар Ю.М.

АНОТАЦІЯ

Метою даної роботи є налагодження системи інформаційного середовища кафедри АПЕПС в системі Microsoft Office 365, використовуючи існуючі частини і додавання власних частин інтерфейсу, аналіз способів налагодження системи, переваг та недоліків цих способів, вибір способу, вивчення та аналіз доступних інструментів, реалізація та розгорнення системи у хмарному середовищі.

Під час виконання дипломної роботи було розглянуто переваги і недоліки вже існуючих аналогів системи, спроектовано і розроблено інформаційну систему кафедри АПЕПС, налагоджено існуючі додатки для керування обліковими записами в системі Office 365 та інтегровано їх в інформаційну систему.

Записка містить 60 сторінок, 54 рисунки, 6 додатків і 13 посилань.

Ключові слова: інформаційна система кафедри, Microsoft Office 365, SharePoint.

ABSTRACT

The purpose of this work is to set up information environment of the department of APEPS in the Microsoft Office 365 system using existing parts and adding own parts of the interface, determine the methods to create this interface, analyze the advantages and disadvantages of these methods, choose appropriate method, study and analyze the available tools, implement and deploy the system to a cloud environment.

During the graduation work the advantages and disadvantages of already existing analogues of the system were considered, the information system of the APEPS department was designed and developed, the applications to manage the accounts in the Office 365 system were debugged and integrated into the information system.

The note contains 60 pages, 54 pictures, 6 attachments and 13 links.

Key words: information system of the department, Microsoft Office 365, SharePoint.

ЗМІСТ

ЗМІСТ	5
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП	9
1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА КАФЕДРИ	11
1.1. Інтерфейс адміністратора	11
1.2. Інтерфейс викладача	12
1.3. Інтерфейс студента	12
1.4. Спільна точка доступу до інтерфейсів	13
1.5. Інтерфейс вікі-сторінки	13
2. АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ТА ОРГАНІЗАЦІЇ СИСТЕМИ ІНФОРМАЦІЙНОГО УЧБОВОГО SEREDOVISHCHA	14
2.1. Система ODEMS	14
2.2. Moodle	16
2.3. Висновки до розділу	18
3. ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОГО SEREDOVISHCHA КАФЕДРИ НА БАЗІ OFFICE 365	19
3.1. Інтегроване середовище розробки Visual Studio Code	19
3.2. Технології розробки	21
3.2.1. Платформа Microsoft Office 365	21
3.2.2. Платформа Microsoft SharePoint Online	22
3.2.3. Мова Javascript	24
3.2.4. Мова Typescript	25
3.2.5. Бібліотека React	26
3.3. Висновки до розділу	27
4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	29

4.1.	Microsoft Graph	30
4.2.	Azure Active Directory	30
4.3.	Карта переходів по додатку	31
4.3.1.	Головна сторінка кафедри АПЕПС	31
4.3.2.	Сторінка адміністрування	33
4.3.3.	Сторінка викладача	33
4.3.4.	Сторінка студента	34
4.4.	Керування групами і користувачами	35
4.5.	Висновки до розділу	37
5.	МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ	
	38	
5.1.	Авторизація користувача в Office 365	38
5.2.	Інтерфейс головної сторінки кафедри АПЕПС	40
5.2.1.	Новини кафедри	40
5.2.2.	Корисні посилання	41
5.2.3.	Сторінки конкретних користувачів	41
5.3.	Сторінка студентів	42
5.4.	Сторінка викладача	44
5.4.1.	Організація своєї роботи	44
5.4.2.	Організація роботи зі студентами	47
5.4.3.	Організовувати роботу групи викладачів	51
5.4.4.	Інтерфейс адміністратора	57
5.5.	Вікі-сторінка	59
5.6.	Висновки до розділу	60
	ВИСНОВКИ	61
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
	ДОДАТОК А	64
	ДОДАТОК Б	66

ДОДАТОК В	73
ДОДАТОК Г	75
ДОДАТОК Д	85
ДОДАТОК Е	87

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface.

Azure AD – Azure Active Directory.

XML – eXtensible Markup Language.

DOM – Document Object Model.

ВСТУП

На сучасному етапі розвитку системи електронного навчання і дистанційна освіта стають все більш затребуваними і актуальними. Електронна освіта має ряд переваг в порівнянні з традиційною формою навчання. Тим не менше, розвиток дистанційної електронної освіти в українських вишах проходить з певними труднощами і перешкодами.

Системи електронного навчання мають ряд переваг порівняно з традиційними методами. До них можна віднести наступні:

- можливість дистанційної взаємодії з викладачем;
- цілодобовий доступ до актуальних матеріалів;
- самостійна робота з електронними матеріалами, з використанням персонального комп'ютера, ноутбука або смартфона;
- можливість вибору місця і часу отримання знань.

Проте виникає необхідність вирішення таких проблем, як: освоєння і популяризація інноваційних педагогічних технологій, передача їх студентам; розробка нових стандартів якості, які б можливо було застосувати до електронних навчальних матеріалів; формування та підвищення інформаційної культури викладачів й оволодіння ними сучасними інформаційними технологіями, підвищення ефективності їх професійної діяльності; створення насиченого сучасного електронного контенту, тобто створення комплексу навчальних матеріалів, придатних для застосування в електронному вигляді.

Однією з проблем при організації системи електронного навчання в умовах кафедри університету є поєднання декількох розроблених додатків (таких як управління користувачами або робочий простір викладача) в одне інформаційне середовище. Платформа Office 365 є однією з найпопулярніших, що використовуються для побудови систем, що вирішують проблеми взаємодії різних груп людей в різноманітних організаціях і бізнес-структурах. Ця платформа надає

базу для вирішення проблем обміну інформацією, файлами, організацією зустрічей, нарад, автоматизації повідомлень та інше.

На сьогодні не існує уніфікованого інтерфейсу середовища кафедри університету.

Метою даної роботи є налагодження системи інформаційного середовища кафедри АПЕПС в системі Microsoft Office 365, використовуючи існуючі частини і додавання власних частин інтерфейсу, аналіз способів налагодження системи, переваг та недоліків цих способів, вибір способу, вивчення та аналіз доступних інструментів, реалізація та розгорнення системи у хмарному середовищі.

В розділі «Постановка задачі розробки інформаційного середовища кафедри» буде описано призначення програмного забезпечення, яке розробляється, його потенційні користувачі. В «Аналізі підходів до створення та організації системи інформаційного учбового середовища» буде проведено аналіз існуючих аналогічних програмних систем. У розділі «Засоби розробки інформаційного середовища на базі Office 365» буде описано платформи та інструментарій що використовується при розробці програмного забезпечення. У «Описі програмної реалізації» буде описано структуру проекту. У розділі «Методика роботи користувача з програмною системою» буде детально описано функціонал системи. У розділі «Висновки» будуть підбиті підсумки по проведених роботах. У «Списку використаних джерел» можна знайти посилання на корисну літературу.

1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА КАФЕДРИ

На сьогодні немає єдиного шаблону для реалізації даної системи. Вихідний продукт повинен представляти собою зручний і нативний інтерфейс для взаємодії і роботи груп користувачів . Потенційно, користувачів можна розділити на декілька груп:

- Студенти;
- Викладачі;
- Адміністратори.

Такий розподіл дозволить встановити базові правила для забезпечення безпеки даних кожної групи і можливість розподілу необхідних прав доступу для даних і ресурсів в середовищі.

1.1. Інтерфейс адміністратора

Однією з найважливіших функцій системи має бути внесення в систему облікових записів студентів та розподіл їх по групах згідно розподілу на кафедрі. Звичайно ж, такі права доступу не можуть мати усі користувачі. Таку можливість було б логічно надати тільки адміністраторам системи. Інша функція системи – управління доступом до цих облікових записів. В разі, наприклад, втрати пароля користувачем, він повинен мати можливість зв'язатися з відповідальною людиною (фізично, наприклад) для допомоги у вирішенні проблеми такого роду. Бувають і інші потреби, з точки зору керування користувачами, які необхідно задовольнити. Наприклад – зміна електронної скриньки користувача, або примусовий вихід з

системи користувача. Не менш важливо – додавати інших адміністраторів до системи, оскільки надавати доступ до одного облікового запису двом людям – погана практика.

1.2. Інтерфейс викладача

Організація групи викладачів потребує найбільшої уваги. Саме викладачі кафедри – ті хто, буде проводити з системою більше всього часу.

Керування своїм розкладом – важлива складова робочого процесу. Тож корисною функцією була б можливість організовувати зустрічі з іншими викладачами та відображати свою зайнятість в своєму календарі так, щоб вона була доступна іншим для перегляду. Доступ до поштової скриньки, основного інструменту комунікації в системі – річ необхідна для ефективної роботи.

Окрім організації своєї роботи, інтерфейс викладача має надавати можливість роботи зі студентами – як зі всією групою (поділитися матеріалами дисципліни, наприклад), так і з кожним студентом окремо (домашні завдання, особисті тести, додаткові матеріали, які студент попросив).

1.3. Інтерфейс студента

З постійною періодичністю виникає потреба надати студентам інформацію. Інформація про дипломну практику, певні вказівки по взаємодії з деканатом, документація, згідно якої треба оформити документи, дати подачі документів на зно, і т.п. – приклади таких потреб. Тож одна з важливих функцій – створити простір, куди можна завантажувати файли з інформацією, де можна переглядати їх та звідки можна завантажувати на локальний носій цю інформацію. Такий підхід збереже час кожного студента і викладача.

Одна з потреб студентів на 4 курсі – знайти дипломного керівника та тему для дипломної роботи. Студент не може знати які теми дипломних робіт можна взяти для роботи у того чи іншого викладача. До того ж не завжди є час та можливість

зв'язатися студенту та викладачу, просто щоб дізнатися які теми є в наявності. Тож корисною функцією системи, як для студентів, так і для викладачів, буде інтерфейс, що дозволив би переглядати наявні теми дипломних робіт на кафедрі, бачити чи вільні вони для розробки, дипломного керівника цієї роботи та його електронну скриньку для зв'язку.

1.4. Спільна точка доступу до інтерфейсів

Недостатньо розробити інтерфейси, які будуть працювати. Це не робить їх однією системою. Під системою розуміється набір логічно зв'язаних компонентів. І, в конкретному випадку, потрібно зв'язати вже названі інтерфейси адміністратора, викладача та студента. Для цього потрібно створити інтерфейс, який буде, за наявності прав доступу, надавати доступ до конкретних інтерфейсів. Також цей інтерфейс може надавати доступ до вже існуючих джерел інформації, які можуть знадобитися кожному – такі як розклад пар в інституті, електронний кампус або сайт кафедри.

1.5. Інтерфейс вікі-сторінки

Оскільки система а) досить багатофункціональна б) пропонує новий підхід до навчання, то вона може бути досить незвичною для нового користувача. Для того щоб допомогти користувачу ознайомитися з системою, необхідно розробити функціонал вікі сторінок (сторінок знань), які будуть частиною системи і завжди під рукою у користувача. На вікі сторінці будуть описуватися кроки, які потрібно зробити в системі для використання її функцій.

2. АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ТА ОРГАНІЗАЦІЇ СИСТЕМИ ІНФОРМАЦІЙНОГО УЧБОВОГО СЕРЕДОВИЩА

На сьогодні немає єдиного шаблону для реалізації системи керування обліковими записами на рівні кафедри в середовищі Office 365. Однак, є аналогічний досвід організації роботи учасників навчального процесу в інших спеціально розроблених для цього системах.

2.1. Система ODEMS

Гарним зразком є комплексне рішення ODEMS (англ. Optimal Digital Education Management System). Рішення представляє собою повне автоматизоване програмне забезпечення для управління освітою, підходить для будь-якого навчального закладу. Система надає широкий спектр можливостей (Рис. 2.1), які можна налаштовувати окремо для кожного процесу.

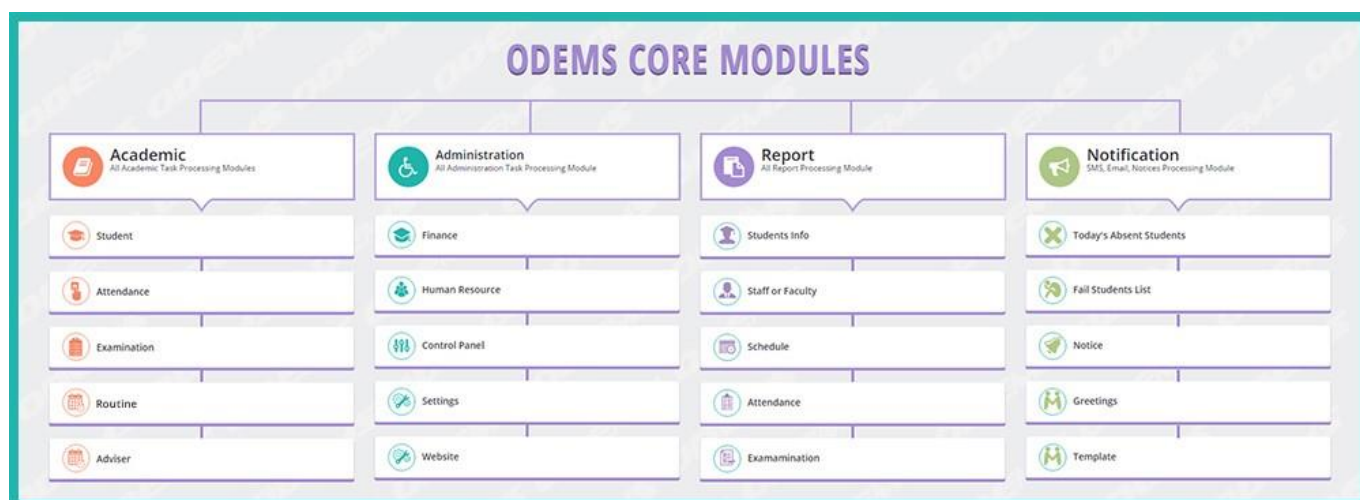


Рисунок 2.1 – Схема роботи ODEMS

Основною перевагою ODEMS є те що здебільшого всі дані зберігаються в «хмарному» сховищі, а тому доступні користувачам у будь-який момент та в будь-якому місці. Також система має окремий модуль майже для кожної задачі (Рис. 2.2), наприклад, студентський модуль для управління обліковими записами студентів та академічний модуль для обліку присутності та успішності.

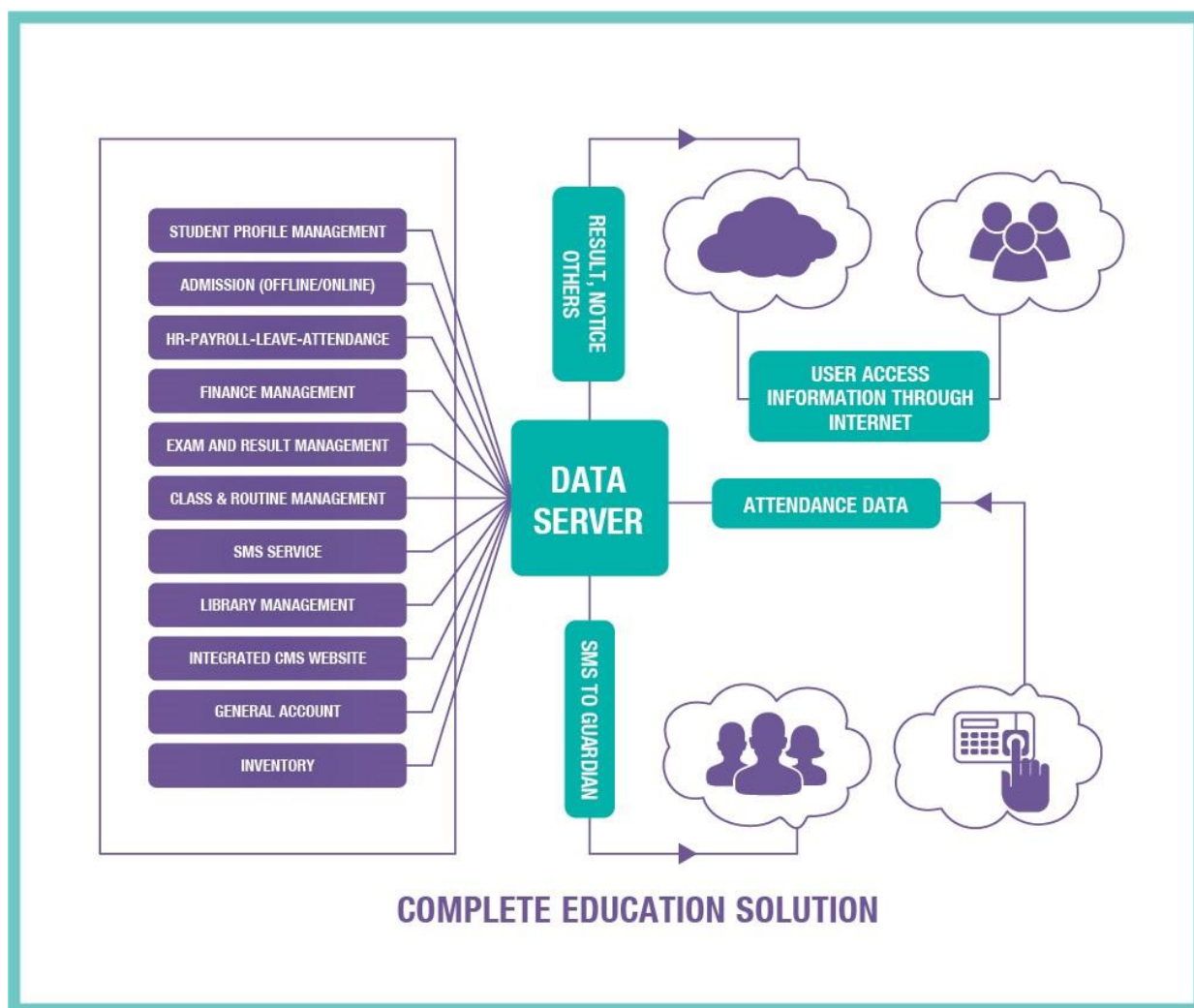


Рисунок 2.2 – Схема роботи ODEMS

2.2. Moodle

Система Moodle має широкий набір функціональності, притаманний платформам електронних систем навчання, системам управління курсами, системам управління навчанням або віртуальним навчальним середовищам. Функціонал Moodle надає можливість викладачам створювати ефективні сайти для онлайн-навчання. Дану систему дистанційного навчання можна використовувати як в навчанні школярів, студентів, так і при підвищенні кваліфікації, бізнес-навчанні.

Типова функціональність Moodle включає [1]:

- здача завдань;
- дискусійні форуми;
- завантаження файлів;
- оцінювання;
- обмін повідомленнями;
- календар подій;
- новини та анонси подій (для різних рівнів: сайт, курс, навчальна група);
- онлайн тестування;
- вікі.

Розробники можуть створювати додаткові модулі з новою додатковою функціональністю. Система Moodle підтримує різні типи модулів:

- типи діяльностей (включаючи можливі навчальні ігри);
- типи ресурсів;
- типи тестових питань;
- типи полів для бази даних (мається на увазі діяльність база даних);
- теми для оформлення;
- різні методи автентифікації;
- різні методи зарахування на курс;
- фільтри для контенту.

Використовуючи цю можливість розроблено багато сторонніх модулів [2].

У середовищі Moodle студенти отримують [1]:

— доступ до навчальних матеріалів (тексти лекцій, завдання до практичних/лабораторних та самостійних робіт; додаткові матеріали (книги, довідники, посібники, методичні розробки) та засобів для спілкування і тестування «24 на 7»;

— засоби для групової роботи (Вікі, форум, чат, семінар, вебінар);

— можливість перегляду результатів проходження дистанційного курсу студентом;

— можливість перегляд результатів проходження тесту;

— можливість спілкування з викладачем через особисті повідомлення, форум, чат;

— можливість завантаження файлів з виконаними завданнями;

— можливість використання нагадувань про події у курсі.

Викладачам надається можливість:

— використання інструментів для розробки авторських дистанційних курсів;

— розміщення навчальних матеріалів (тексти лекцій, завдання до практичних/лабораторних та самостійних робіт; додаткові матеріали (книги, довідники, посібники, методичні розробки) у форматах .doc, .odt, .html, .pdf, а також відео, аудіо і презентаційні матеріали у різних форматах та через додаткові плагіни;

— додавання різноманітних елементів курсу;

— проведення швидкої модифікації навчальних матеріалів;

— використання різних типів тестів;

— автоматичного формування тестів;

— автоматизації процесу перевірки знань, звітів щодо проходження студентами курсу та звітів щодо проходження студентами тестів;

— додавання різноманітних плагінів до курсу дозволяє викладачу використовувати різноманітні сторонні програмні засоби для дистанційного навчання.

2.3. Висновки до розділу

Беручи за приклад ці дві системи, можна побудувати власну інформаційну систему середовища кафедри АПЕПС, в якій буде враховано позитивний досвід обох. Платформа Microsoft Office 365 дозволяє як створювати власний інтерфейс для рішень будь-якого типу, так і надає прикладний програмний інтерфейс для роботи з «хмарними» сховищами для файлів. Використовуючи такі особливості, можна побудувати досить гнучку систему з підтримкою різних типів користувачів, можливістю динамічного наповнення контенту, зручним інтерфейсом обміну інформації (повідомлень, файлів) між користувачами, інтерфейсом організації робочого процесу (календар, органайзер). Також необхідно розробити базу знань для того, щоб новому користувачу задача ознайомлення з системою не була складнішою за користування цією системою. База знань має бути частиною самої системи і має бути розроблена у вигляді набору вікі-сторінок.

3. ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА КАФЕДРИ НА БАЗІ OFFICE 365

На сьогоднішній день навряд чи можна знайти систему, яка буде писатися з повного нуля без сторонніх бібліотек або в блокноті. Для прискорення розробки програмних систем використовуються інтегровані середовища розробки і технології розробки.

3.1. Інтегроване середовище розробки Visual Studio Code

Інтегроване середовище розробки — комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

Інтегровані середовища розробки створені для того, щоб максимізувати продуктивність програміста, надавши йому пов'язані інструменти розробки зі схожими інтерфейсами як одну програму, в якій відбуватиметься весь процес розробки й яка надає необхідні функції для модифікації, компілювання, розгортання та налагодження програмного забезпечення.

Visual Studio Code — засіб для створення, редагування та відлагодження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X (рис. 3.1). Редактор містить вбудований засіб відлагодження, інструменти для роботи з Git і засоби рефакторингу, навігації по коду, автодоповнення і контекстної підказки.

Visual Studio Code дозволяє створювати і підключати сторонні розширення для додавання нової функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду, додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення.

Продукт підтримує розробку для платформ ASP.NET і Node.js, і позиціонується як рішення, що дозволяє обійтися без повного інтегрованого середовища розробки. Серед підтримуваних мов і технологій: JavaScript, C++, C#, TypeScript, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, HTML, CSS, LESS і SASS, Haxe [3].

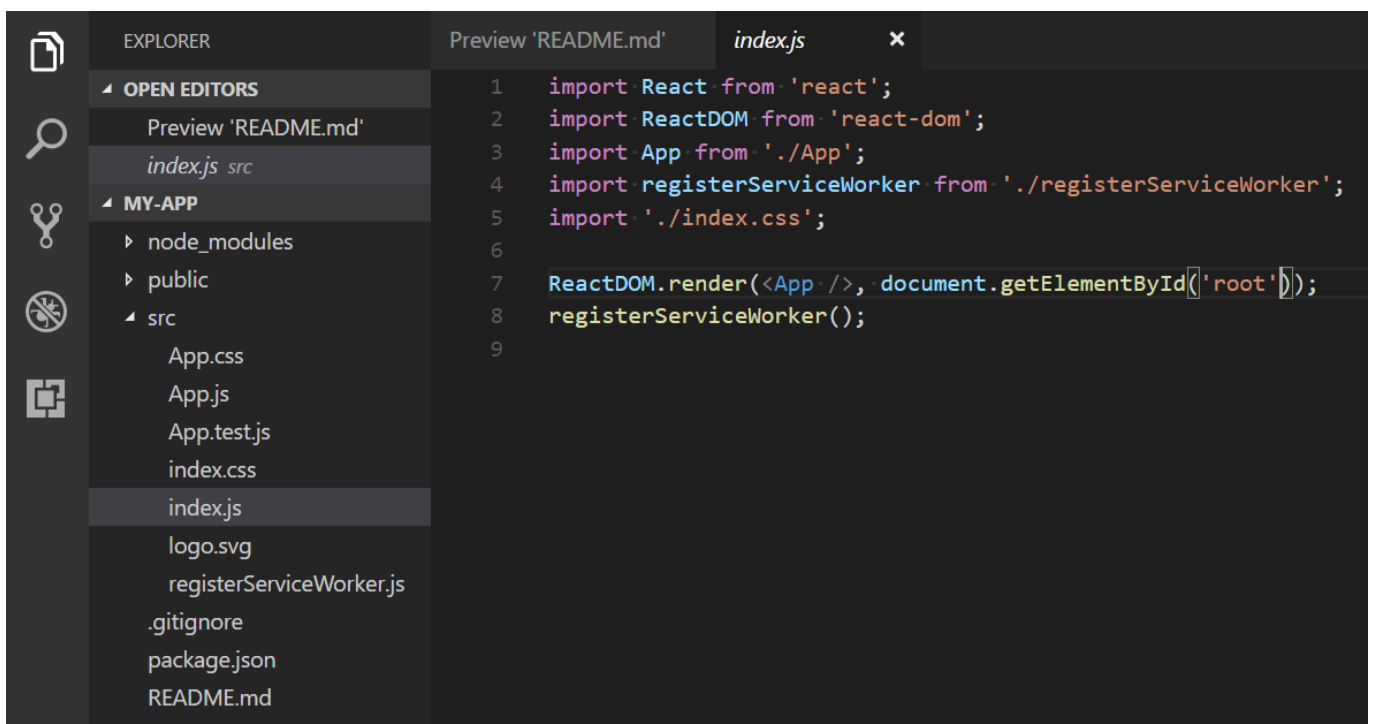


Рисунок 3.1 – Середовище розробки Visual Studio Code

3.2. Технології розробки

Для прискорення розробки програмних систем досить часто використовуються вже створені технології розробки – платформи, бібліотеки, фреймворки. Це не тільки прискорює розробку, а й дозволяє абстрагуватися до рівня вирішення проблеми, а не написання програмного коду.

3.2.1. Платформа Microsoft Office 365

Платформа Microsoft Office 365 – це хмарний власницький інтернет-сервіс і програмне забезпечення компанії Microsoft, що розповсюджується за схемою «програмне забезпечення + послуги» (англ. Software + Services). Хмарний формат означає, що дані зберігаються в центрі обробки даних, а не на комп'ютері, що забезпечить користувачам доступ до документів і даних через браузер з різних пристроїв з можливістю виходу в Інтернет.

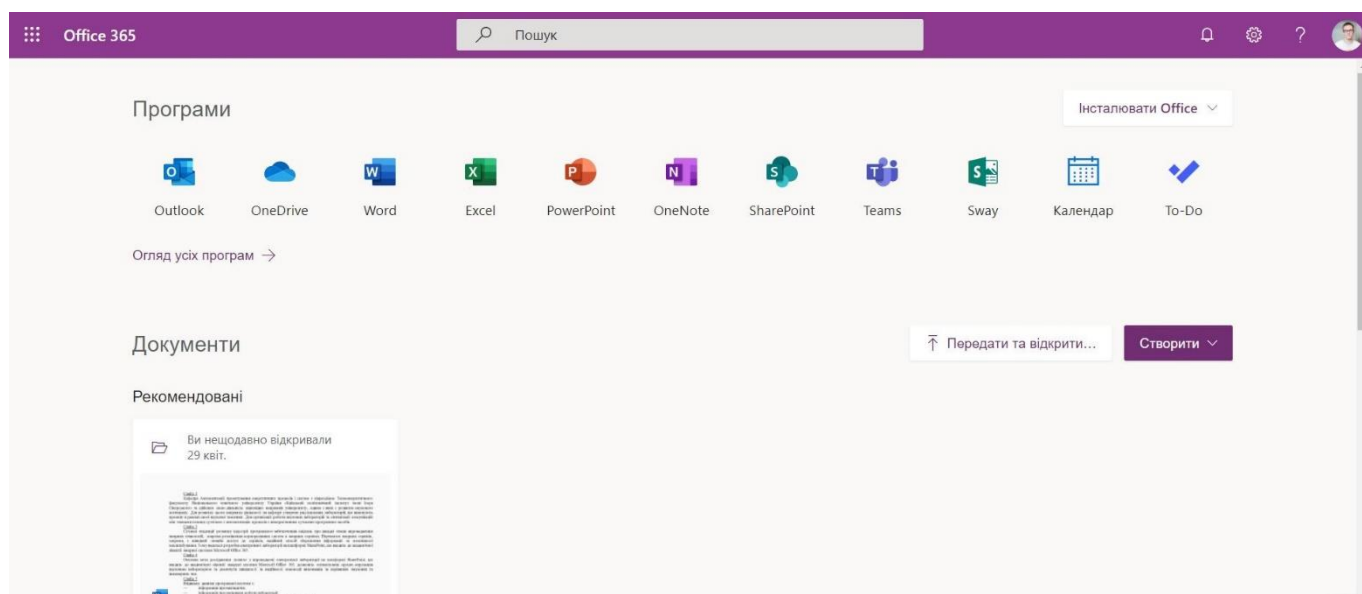


Рисунок 3.2 – Середовище Office 365

Хмарний базовий пакет послуг Microsoft Office 365 включає в себе:

- електронна пошта бізнес-класу на сервері Exchange;
- портал SharePoint, що дозволяє розміщувати публічні і приватні сайти та веб-додатки, надавати доступ до спільної інформації;
- комунікатор Lync, що дозволяє не тільки обмінюватися текстовими фразами, але і проводити відео та аудіо конференції, а також проводити показ робочого столу;
- додатки останньої версії Microsoft Office;
- місце в OneDrive.

Також Office 365 містить численну кількість інструментів, що дозволяють розширити можливості взаємодії учасників навчального процесу.

3.2.2. Платформа Microsoft SharePoint Online

Платформа Microsoft SharePoint Online дозволяє організаціям з легкістю створювати налаштовувані сайти та додатки для роботи в командах, групах та керувати цими сайтами [4]. Крім того, є можливість розгортки у внутрішній мережі порталу організації для розповсюдження інформації та новин по всій організації.

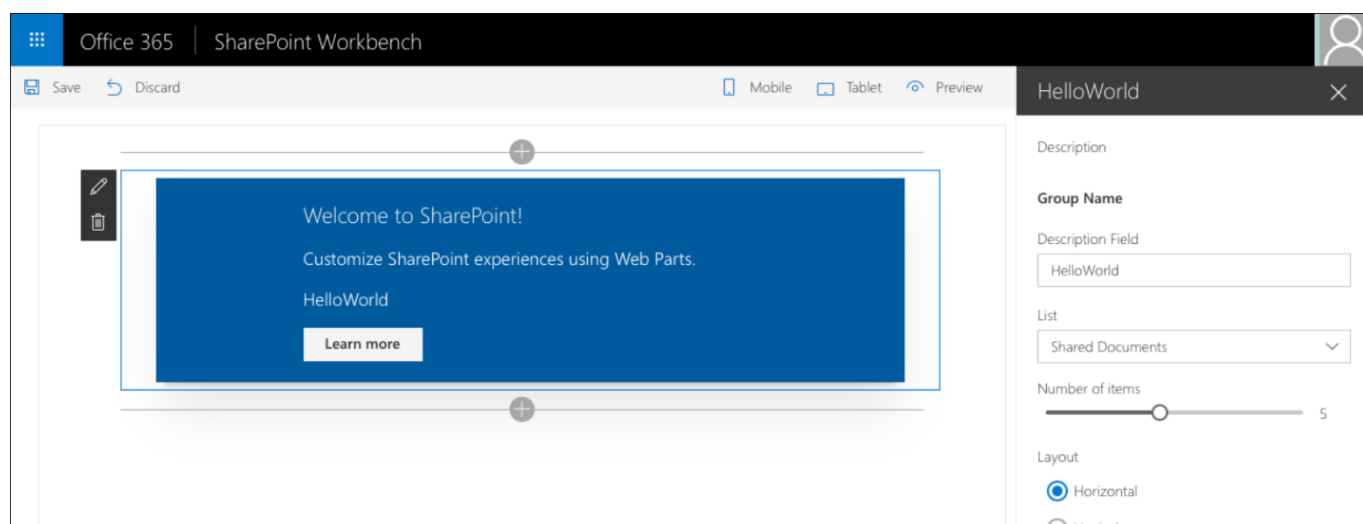


Рисунок 3.3 – Зразок веб-частини SharePoint Framework

Існують різні випуски SharePoint, всі вони мають бути розміщені на сервері SharePoint Server [5]:

- випуск SharePoint Standard, побудований на Microsoft SharePoint Foundation містить базовий пакет ключових можливостей;

- випуск SharePoint Enterprise включає в себе додаткові можливості: покращений пошук, інтеграція с середовищем бізнес-аналітики, інформпанель, PowerPivot та PerformancePoint, Microsoft Office Access, Visio, Excel, SharePoint Enterprise розширення для пошуку;

- випуск SharePoint Online: підтримка розширень на базі SharePoint Framework (рис. 3.3), сучасний дизайн SharePoint UX, інтеграція з Yammer та Office 365 Groups, інтеграція з Outlook Web App, перехід до Apps Concept.

SharePoint Framework дозволяє розробляти додатки використовуючи сучасні технології.

SharePoint Framework (SPFx) – це фреймворк для побудови сторінок та веб-частин, що надає повну підтримку для розробку клієнтських додатків, легку інтеграцію з даними в SharePoint і підтримку інструментів з відкритим кодом [6]. Властивості веб-частин:

- додатки виконуються в контексті конкретного користувача і з'єднання в браузері. Не потрібна підтримка iFrames (JavaScript-код вбудовано в сторінку);

- елементи управління знаходяться в звичайній DOM-структурі сторінки;

- елементи управління по своїй природі адаптивні і легкодоступні;

- не залежить від використовуваного JavaScript фреймворку. Можливе використання будь-якого, наприклад React, Handlebars, Knockout, Angular та інші;

- розробка базується на таких загальнодоступних інструментах npm, TypeScript, Yeoman, webpack, і gulp;

- можливість розробника отримати доступ до життєвого циклу додатку;

- кінцеві користувачі можуть використовувати клієнтські додатки що одобрені адміністраторами порталу (або особами з делегованими повноваженнями).

Історично склалося, що веб-частини як збірки C #, встановлені на хмарних серверах. Тим не менше, сучасні моделі розробки в більшості випадків включають в себе JavaScript, що виконується в браузері, що робить виклики REST API до сервісів SharePoint та Office 365.

3.2.3. Мова Javascript

Мова JavaScript (JS) — динамічна, об'єктно-орієнтована мова програмування. JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам (наприклад, функції як об'єкти першого класу, об'єкти як списки, каррування, анонімні функції, замикання), що додає мові додаткову гнучкість.

Поряд з HTML та CSS, JavaScript є однією з трьох основних технологій Всесвітньої павутини. [7] JavaScript надає інтерактивні веб-сторінки та, таким чином, є невід'ємною частиною веб-додатків. Переважна більшість веб-сайтів використовують її, [8] і всі основні веб-переглядачі мають спеціальний рушій JavaScript для її виконання.

Мова JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне приведення типів; автоматичне прибирання сміття;
- анонімні функції.

Як мультпарадигменна мова програмування, JavaScript підтримує такі стилі програмування, як функціональний та імперативний (включаючи об'єктно-орієнтований та прототипо-орієнтований). Вона має API для роботи з текстом, масивами, датами, регулярними виразами та базовими маніпулюваннями DOM (Document Object Model), але сама мова не включає в себе ніяких засобів для введення/виведення (наприклад, для мережевої взаємодії, зберігання інформації та графічної взаємодії), так як спирається на середовище, в якому вона вбудована.

Спочатку інтерпретатори JavaScript були реалізовані лише на стороні клієнта в веб-переглядачах. Тепер двигуни JavaScript вбудовані в багато інших типів програмного забезпечення, включаючи серверну сторону в веб-серверах та базах даних, а також в не веб-орієнтованих програмах, таких як текстові редактори та програмне забезпечення PDF, а також у режимі середовища виконання, що робить JavaScript доступною мовою для написання мобільних програм.

3.2.4. Мова Typescript

TypeScript — мова програмування, представлена Microsoft восени 2012; позиціонується як засіб розробки веб-застосунків, що розширює можливості JavaScript.

Розробником мови TypeScript є Андерс Гейлсберг (англ. Anders Hejlsberg), який створив раніше C#, Turbo Pascal і Delphi.

Код експериментального компілятора, котрий трансліює код TypeScript в представлення JavaScript, поширюється під ліцензією Apache, розробка ведеться в публічному репозиторії через сервіс CodePlex. Специфікації мови відкриті і опубліковані в рамках угоди Open Web Foundation Specification Agreement (OWFa 1.0).

TypeScript є зворотно сумісним з JavaScript. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою Node.js.

Переваги над JavaScript:

- можливість явного визначення типів (статична типізація),
- підтримка використання повноцінних класів (як в традиційних об'єктно-орієнтованих мовах),
- підтримка підключення модулів.

За задумом ці нововведення мають підвищити швидкість розробки, прочитність, рефакторинг і повторне використання коду, здійснювати пошук помилок на етапі розробки та компіляції, а також швидкодію програм.

Планується, що в силу повної зворотної сумісності адаптація існуючих застосунків на нову мову програмування може відбуватися поетапно, шляхом поступового визначення типів. Підтримка динамічної типізації зберігається — компілятор TypeScript успішно обробить і не модифікований код на JavaScript. Основний принцип мови — весь існуючий код на JavaScript сумісний з TypeScript, тобто в програмах на TypeScript можна використовувати стандартні JavaScript-бібліотеки і раніше створені напрацювання. Більш того, можна залишити існуючі JavaScript-проекти в незмінному вигляді, а дані про типізації розмістити у вигляді анотацій, які можна помістити в окремі файли, які не заважатимуть розробці і прямому використанню проекту (наприклад, подібний підхід зручний при розробці JavaScript-бібліотек).

На момент релізу представлені файли для сприйняття розширеного синтаксису TypeScript для Vim і Emacs, а також плагін для Microsoft Visual Studio.

Одночасно з виходом специфікації розробники підготували файли з деклараціями статичних типів для деяких популярних JavaScript-бібліотек, серед яких jQuery.

3.2.5. Бібліотека React

React — JavaScript бібліотека з відкритим кодом для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-

сторінки, з якими стикаються в розробці односторінкових додатків. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників.

React дозволяє розробникам створювати великі веб-додатки, які використовують дані і можуть змінюватися з часом без перезавантаження сторінки. Він спрямований перш за все на забезпечення швидкості, простоти та масштабованості. React працює лише з користувацькому інтерфейсами у програмах. Це відповідає View у моделі Model-View-Controller (MVC), і може використовуватися в комбінації з іншими бібліотеками JavaScript або фреймворками у MVC, наприклад AngularJS.

Ключові особливості React:

— компоненти — дозволяють розробити користувацький інтерфейс, використовуючи незалежні, багаторазові частини, і розглядати кожну з цих частин окремо

— властивості — передаються компоненту з батьківського компонента. Компоненти отримують властивості як єдиний набір незмінних значень [9] (об'єкт JavaScript). Кожного разу, коли змінюється значення параметра, викликається функція відображення компонента, що дозволяє компоненту відобразити зміну. віртуальна об'єктна модель документу або віртуальна DOM (рисунок 3.7). React створює кеш даних у пам'яті, обчислює отримані зміни, а потім ефективно оновлює відображений DOM в веб-переглядачі [10]. Це дозволяє програмісту писати код так, ніби вся сторінка оновлюється при кожній зміні, а бібліотека React оновлює лише ті підкомпоненти, які дійсно змінилися.

3.3. Висновки до розділу

В розділі наведено перелік використаних технологій, а саме: інтегроване середовище розробки Visual Studio Code, . платформа Microsoft Office 365, платформа

Microsoft SharePoint Online, бібліотека React, мови Javascript та Typescript. Також було приведено стислу характеристику кожної технології.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Програмна реалізація являє собою зв'язку з 1) набору веб-частин на базі SharePoint Framework, що реалізовані за подібною структурою (рис. 4.1) 2) набору сайтів створених за допомогою SharePoint Online.

Шар компонентів відповідає за частини інтерфейсу, шар сервісів – за зв'язок з Microsoft Graph та Azure Active Directory, а шар моделей – за представлення даних.

Далі буде детальніше розглянуто окремі шари приведеної вище структури та карту переходів додатку.



Рисунок 4.1 – Структура веб-частин

4.1. Microsoft Graph

Microsoft Graph є провайдером даних та інформації в Microsoft 365. Microsoft Graph забезпечує уніфікований інтерфейс, який використовується для доступу до більшості даних у службах Office 365 [11].

Microsoft Graph представляє API для:

- Azure Active Directory;
- служби мобільності та безпеки підприємства: Identity Manager, Intune, Advanced Threat Analytics;
- служби Office 365: SharePoint, OneDrive, Outlook / Exchange, Microsoft Teams, OneNote, Planner та Excel;
- служби Windows 10.

Microsoft Graph пов'язує ресурси між цими службами, використовуючи зв'язки. Наприклад, користувач і група за допомогою відносини “memberOf”.

4.2. Azure Active Directory

Office 365 використовує Azure Active Directory (Azure AD) для забезпечення доступу до своїх API, доступ до яких здійснюється через Microsoft Graph. Azure AD використовує протокол авторизації OAuth. Коли додаток завершує потоки авторизації OAuth, він отримує тимчасовий токен доступу (рисунок 4.2). Токен забезпечує доступ до певних ресурсів від імені користувача, використовуючи дозволи, надані додатку цим користувачем [12].

Для того щоб додаток мав доступ до сторонніх API, він повинен бути зареєстрований як програма в Azure AD. У процесі реєстрації зазначається ряд параметрів, таких як URL-адреса, де розміщується додаток, ресурси, для яких програма потребує доступу, і т.е. (Рис. 4.2).

Після реєстрації додатку в Azure AD, і використання наданого йому унікального ідентифікатора в маніфесті додатку, отриманий додатком токен після процесу

авторизації матиме необхідні дозволи для доступу до вказаних при реєстрації API [13].



Рисунок 4.2 – Схема автентифікації для використання сторонніх API

4.3. Карта переходів по додатку

В процесі створення додатку була поставлена задача: досягти комфорту у використанні системи. Для системи з насиченим функціоналом важливо врахувати не тільки роботу з кожним окремим інтерфейсом, а і зручність досягнення того чи іншого інтерфейса. В цьому пункті буде описано карту переходів по додатку.

4.3.1. Головна сторінка кафедри АПЕПС

Головна сторінка кафедри являє собою єдину точку доступу до системи (Рис. 4.3). З неї можна перейти на:

— Сторінки, що не є частиною системи (сайт розкладу НТУУ КПІ, функціонуючий сайт кафедри АПЕПС, електронний кампус НТУУ КПІ)

- сторінку адміністрування;
- сторінку студента;
- сторінку викладача;
- вікі сторінку.



Рисунок 4.3 – Головна сторінка сайту

Таким чином, починаючи з цієї сторінки можна дістатися до будь-якого функціоналу системи. Наявність однієї точки входу в систему є важливим моментом для кінцевого користувача, оскільки йому досить складно одразу запам'ятовувати велику кількість інформації. Перевантаження інформацією може призвести до того, що користувач припинить користуватися системою, і повернеться до традиційної системи організації учбового процесу.

4.3.2. Сторінка адміністрування

Сторінка адміністратора містить інтерфейси для керування користувачами з точки зору системи Office 365 і з точки зору кафедри (Рис. 4.4).



Рисунок 4.4 – Сторінка адміністратора

Також адміністратор має доступ до своїх пошти, календаря та документів.

4.3.3. Сторінка викладача

Сторінка викладача надає досить багатий функціонал для роботи (Рис. 4.5). Він включає в себе:

— календар, пошта і документи групи;

- особисті календар, пошту і документи;
- роботу з учбовими просторами (організація роботи студентських груп);
- робота з подіями (можливість організувати подію на групу викладачів).



Рисунок 4.5 – Сторінка викладача

Таким чином цей інтерфейс може бути основним місцем роботи викладача за персональним комп'ютером.

4.3.4. Сторінка студента

Сторінка студента надає досить скромний, але тим не менш корисний, функціонал (Рис. 4.6).

- перегляд тем актуальних дипломних робіт;
- інформація з переддипломної практики.

Системою передбачено додання нової інформації на цій сторінці.

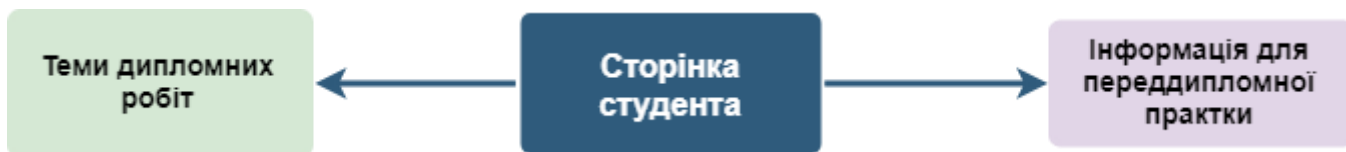


Рисунок 4.6 – Сторінка студента

Ця сторінка носить інформативний характер для студентів. Припускається, що саме на цій сторінці будуть викладатися корисні студентам матеріали. Таким чином, усю необхідну інформацію і документи можна буде знайти тут.

4.4. Керування групами і користувачами

Стандартний функціонал Office 365 не надає зручного інтерфейсу для опрацювання набору студентів – внесення користувачів до групи з наданням необхідних прав доступу, тож було розроблено і вбудовано в систему інтерфейс (Рис. 4.7), який задовольнить ці потреби.

Групи Користувачі

Додати групу Показ груп Додати користувачів до групи

Назва групи *

Опис групи *

Префікс електронної пошти *

@kpiapeps.onmicrosoft.com

Додати групу

Рисунок 4.7 – Веб-частина для керування

Цей інтерфейс надає як стандартні можливості перегляду учасників системи (Рис. 4.8), так і додавання нових учасників, імпортуючи їх з файлів формату Excel (Рис. 4.9).

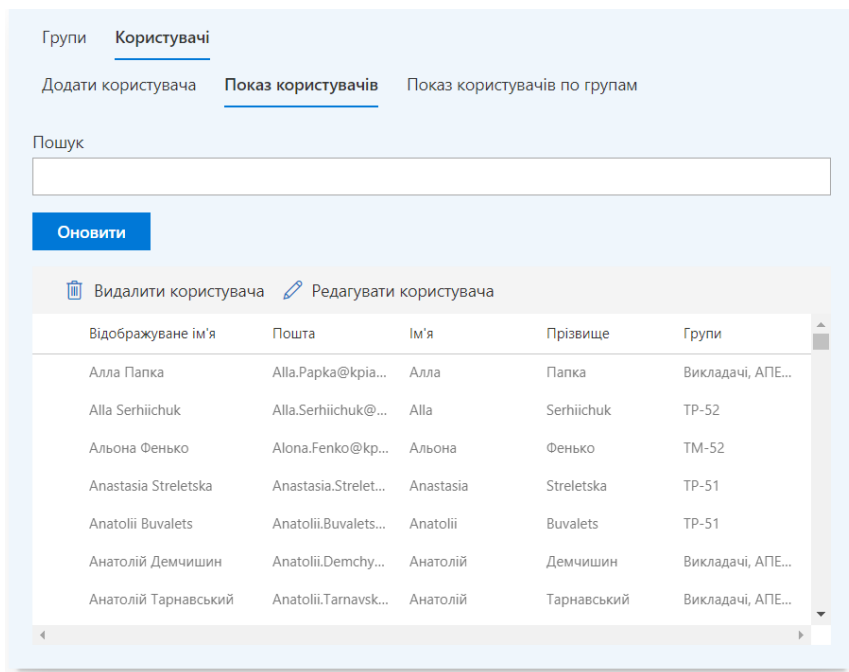


Рисунок 4.8 – Інтерфейс перегляду учасників системи

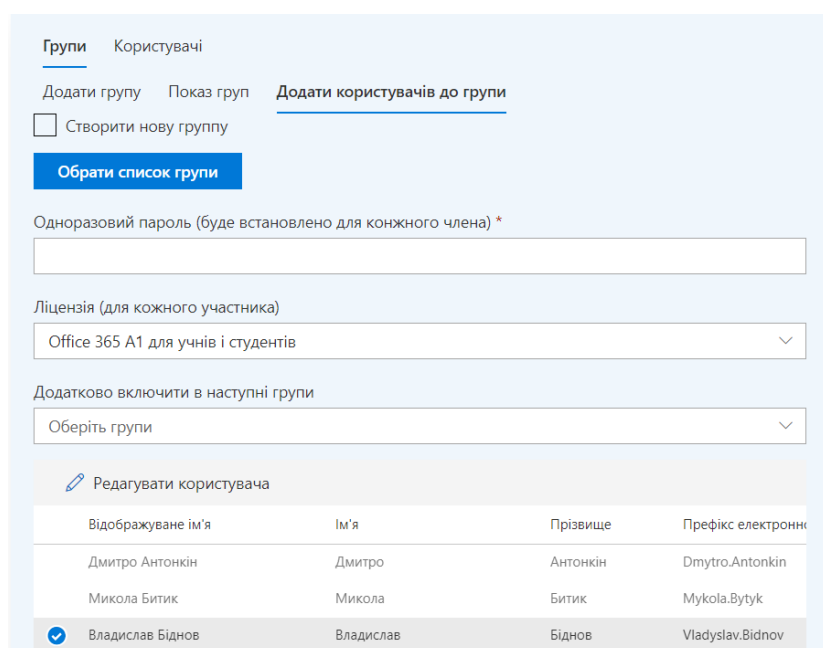


Рисунок 4.9 – Інтерфейс додавання учасників до групи

Далі наведено програмний код створення нової групи в системі (Рис. 4.10).

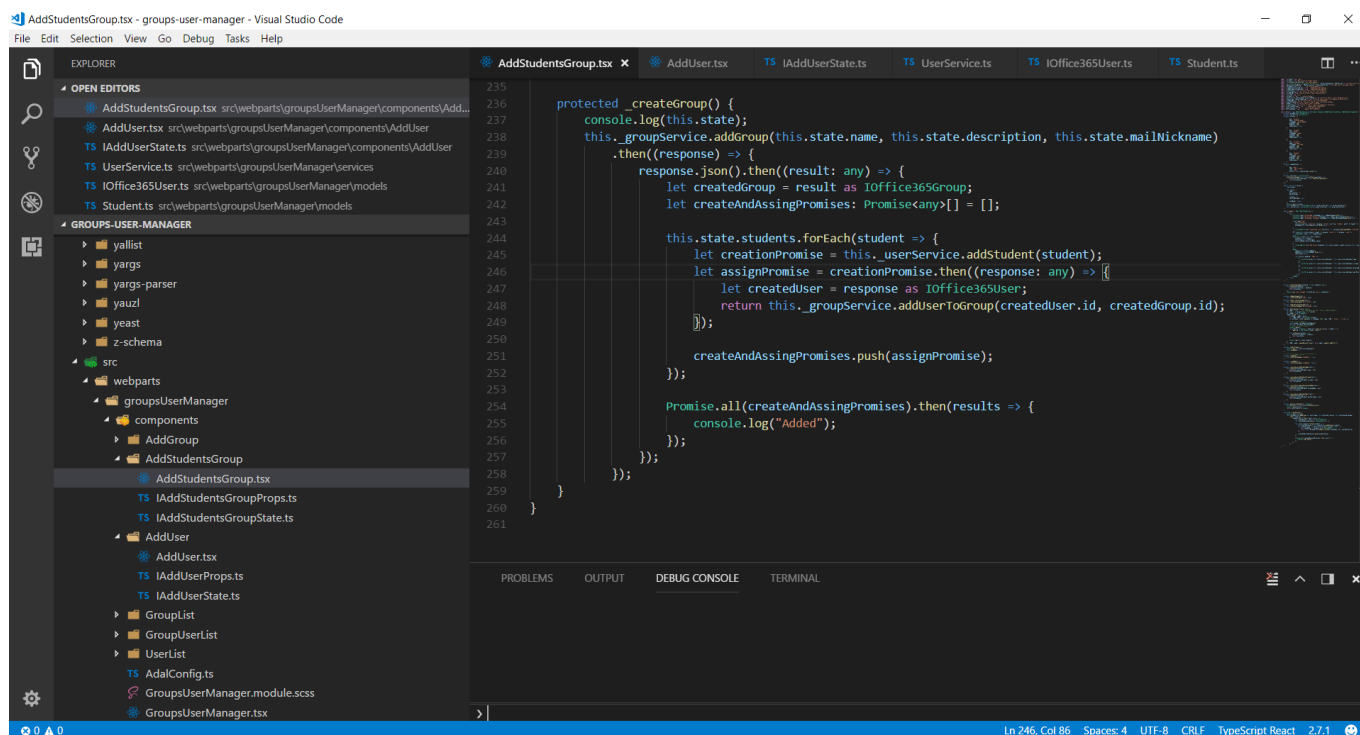


Рисунок 4.10 – Інтерфейс додавання учасників до групи

Інші можливості цього інтерфейсу – перегляд існуючих груп і перегляд користувачів по групах. Цей функціонал аналогічний до того що надає стандартний інтерфейс Office та розроблений як частина додатку для зручності користувача.

4.5. Висновки до розділу

У цьому розділі було описано використання конкретних технологій Microsoft Office 365, які було застосовано при розробці програмного продукту – Microsoft Graph API та Azure AD. Було наведено розроблену карту додатку, а також часткову реалізацію веб-частини для керування групами і користувачами.

5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

У даному розділі буде розглянуто методику роботи користувача з програмною системою. Спочатку буде розглянуто авторизацію користувача в системі, потім буде розглянуто взаємодію з різними інтерфейсами.

5.1. Авторизація користувача в Office 365

Для того щоб авторизуватися в системі, для початку, необхідно мати обліковий запис Office 365 на домені кафедри АПЕПС – kpiarpeps.onmicrosoft.com. Обліковий запис конкретного користувача, скоріш за все, вже створено – цю інформацію можна уточнити у адміністратора(-ів) системи.

Далі необхідно зайти на веб-сайт <https://kpiarpeps.sharepoint.com/sites/ApepsPage>. Це сторінка кафедри. На цьому етапі, якщо ви попередньо не авторизовані в системі, вас буде перенаправлено на сторінку авторизації (Рис. 5.1).

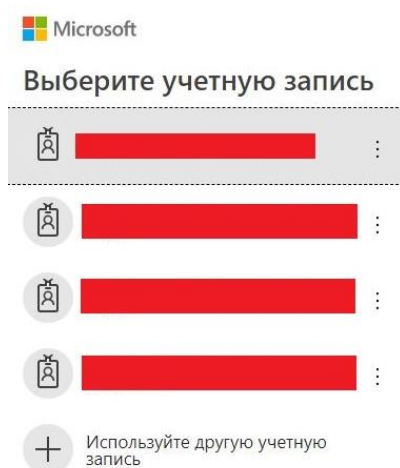
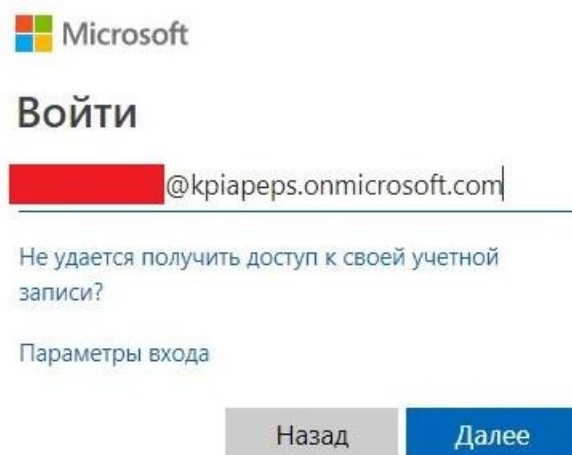


Рисунок 5.1 – Сторінка авторизації

Користувачу буде запропоновано обрати вже існуючий обліковий запис, або використати інший обліковий запис. Нового користувача цікавить другий варіант.

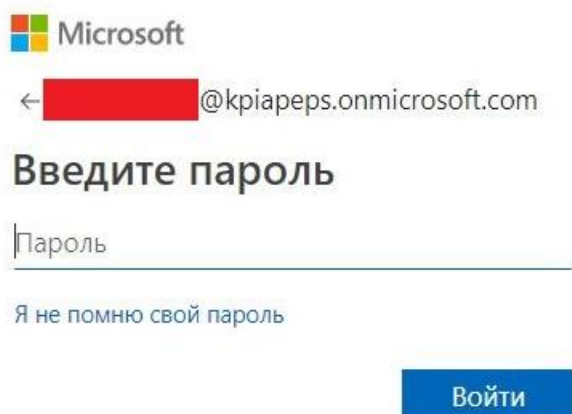
Після того як користувач натискає кнопку «Використати інший обліковий запис», його буде направлено на сторінку з формою для вводу облікового запису (Рис. 5.2).



The screenshot shows the Microsoft login interface. At the top is the Microsoft logo. Below it is the heading 'Войти' (Sign in). There is a text input field containing a redacted email address followed by '@kpiapeps.onmicrosoft.com'. Below the input field is a link that says 'Не удается получить доступ к своей учетной записи?' (Having trouble signing in?). Below that is a link 'Параметры входа' (Sign-in options). At the bottom are two buttons: 'Назад' (Back) and 'Далее' (Next).

Рисунок 5.2 – Форма для вводу облікового запису

Після того як користувач вводить свій обліковий запис, він натискає синю кнопку «Далі» і переходить на сторінку з формою для вводу паролю.



The screenshot shows the Microsoft login interface for password entry. At the top is the Microsoft logo. Below it is a back arrow and a text input field containing a redacted email address followed by '@kpiapeps.onmicrosoft.com'. Below the input field is the heading 'Введите пароль' (Enter password). There is a text input field for the password with the placeholder text 'Пароль'. Below the input field is a link that says 'Я не помню свой пароль' (I forgot my password). At the bottom right is a blue button labeled 'Войти' (Sign in).

Рисунок 5.3 – Форма для вводу облікового пароля

Після вводу паролю користувач натискає синю кнопку «Увійти» і переходить на сторінку кафедри.

Якщо на етапі вводу облікового запису або пароля користувач зробить помилку, система виведе повідомлення про помилку і попросить ввести дані заново.

5.2. Інтерфейс головної сторінки кафедри АПЕПС

Мета головної сторінки кафедри – надавати швидкий і практичний доступ до корисних ресурсів для роботи. Це вхідна в систему точка, з якої можна знайти будь-яку потрібну інформацію.

5.2.1. Новини кафедри

На цій сторінці можна переглянути останні новини кафедри. Для цього є спеціальна карусель новин (Рис. 5.4). На каруселі відображаються 4 останні новини. Можна побачити новини за весь час, натиснувши кнопку «Переглянути все» у правому верхньому кутку. Адміністратор може редагувати цю сторінку і додавати новини, за допомогою кнопки «Додати» в верхньому лівому кутку елемента і надавши посилання на новину, або створивши нову сторінку сайту з новиною.

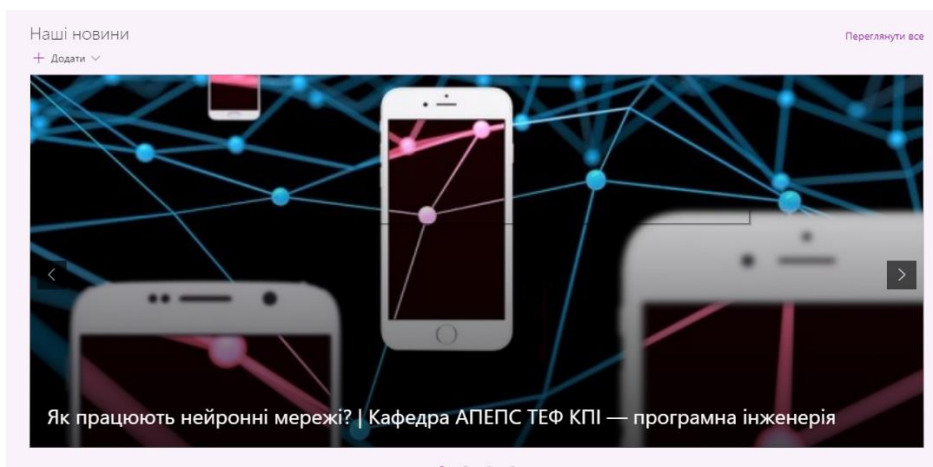


Рисунок 5.4 – Карусель новин

5.2.2. Корисні посилання

Під каруселлю новин можна побачити набір посилань на сторінки (Рис. 5.5), які можуть використовуватися учасниками досить часто. Це такі сторінки:

- сторінка кафедри АПЕПС;
- сторінка розкладу НТУУ «КПІ»;
- сторінка електронного Кампусу НТУУ «КПІ»;
- сторінка допомоги користувачу.

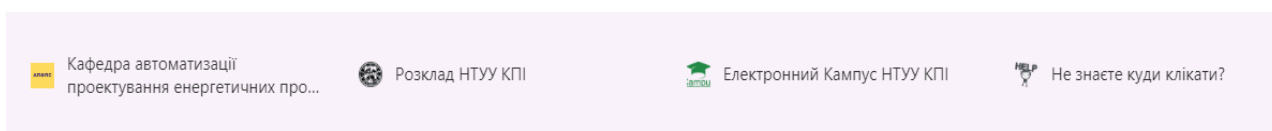


Рисунок 5.5– Набір посилань на часті сторінки

Ці посилання, окрім сторінки допомоги користувачу, не відносяться до ресурсів додатку, але посилання на них має бути, оскільки мета головної сторінки кафедри, як вже зазначалося, – надавати швидкий і практичний доступ до корисних ресурсів для роботи.

5.2.3. Сторінки конкретних користувачів

Під корисними посиланнями можна побачити посилання на сторінки конкретних користувачів (Рис. 5.6). Це такі сторінки:

- сторінка студентів
- сторінка викладачів
- сторінка адміністратора(-ів)

Ці сторінки надають специфічну інформацію для кожного типу користувачів. В системі є різні права доступу, тож студенти мають доступ тільки до сторінки студентів, сторінка викладачів недоступна студентам, а сторінка адміністрування доступна тільки користувачам з правами адміністратора.

Яку інформацію шукаєте?



Рисунок 5.6– Набір посилань на сторінки конкретних користувачів

Адміністратор має права доступу для редагування усіх цих сторінок – і конкретних, і сторінки кафедри.

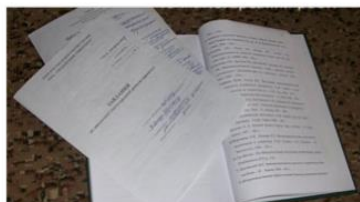
5.3. Сторінка студентів

Головна мета цієї сторінки – надати інформацію, яку частіше за все шукають студенти. Запропонований варіант організації сторінки студентів (Рис. 5.7) – посилання на кожний окремий ресурс. Наразі додано всього два посилання, окрім повернення на головну сторінку: посилання на теми дипломних робіт і посилання на матеріали для переддипломної практики.

На цій сторінці ви знайдете усі корисні посилання для студентів!

[Натисніть тут, щоб повернутися на головну сторінку кафедри](#)

Що вас цікавить?



Теми дипломних робіт



Переддипломна практика

Рисунок 5.7 – Контент сторінки студентів

Посилання «Теми дипломних робіт» веде на відповідну сторінку (Рис. 5.8).

Пошук (Теми дипломних)	+ Створити	Швидке редагування	Експортувати до Excel	Flow	PowerApps	...
Теми дипломних робіт						
Керівник	Email	Тема вільна	Назва теми	+ Додати сто...		
Крамар Ю.М.	yulia.kramar@kpiapeps.onmicrosoft.c...	Ні	Тема 1			
Крамар Ю.М.	yulia.kramar@kpiapeps.onmicrosoft.c...	Так	Тема 2			
Крамар Ю.М.	yulia.kramar@kpiapeps.onmicrosoft.c...	Ні	Тема 3			
Бандурка О.І.	Olena.Bandurka@kpiapeps.onmicros...	Ні	Тема 4			
Бандурка О.І.	Olena.Bandurka@kpiapeps.onmicros...	Так	Тема 5			
Бандурка О.І.	Olena.Bandurka@kpiapeps.onmicros...	Ні	Тема 6			
Бандурка О.І.	Olena.Bandurka@kpiapeps.onmicros...	Так	Тема 7			
Бандурка О.І.	Olena.Bandurka@kpiapeps.onmicros...	Так	Тема 8			
Гайдаржи В.І.	Volodymyr.Haidarzhy@kpiapeps.onm...	Так	Тема 9			
Гайдаржи В.І.	Volodymyr.Haidarzhy@kpiapeps.onm...	Так	Тема 10			
Гайдаржи В.І.	Volodymyr.Haidarzhy@kpiapeps.onm...	Так	Тема 11			

Рисунок 5.8 – Сторінка з списком дипломних робіт

На цій сторінці можна побачити список актуальних дипломних робіт, чи вільні вони, керівників конкретних дипломних робіт та їх пошти. Цей список можна експортувати до Excel. Адміністратор може редагувати цей список. Більш того, він може це робити досить швидко за допомогою швидкого редагування – можна редагувати існуючі та додавати нові рядки використовуючи скопійовані з таблиці записів формату Excel комбінаціями Ctrl+C, Ctrl+V.

Посилання «Переддипломна практика» веде на сторінку з матеріалами для переддипломної практики (Рис. 5.9). На цій сторінці можна знайти перелік документів, які можуть бути корисні студентам 4го курсу.

Адміністратор може оновлювати ці документи, додавати нові та видаляти старі і неактуальні.

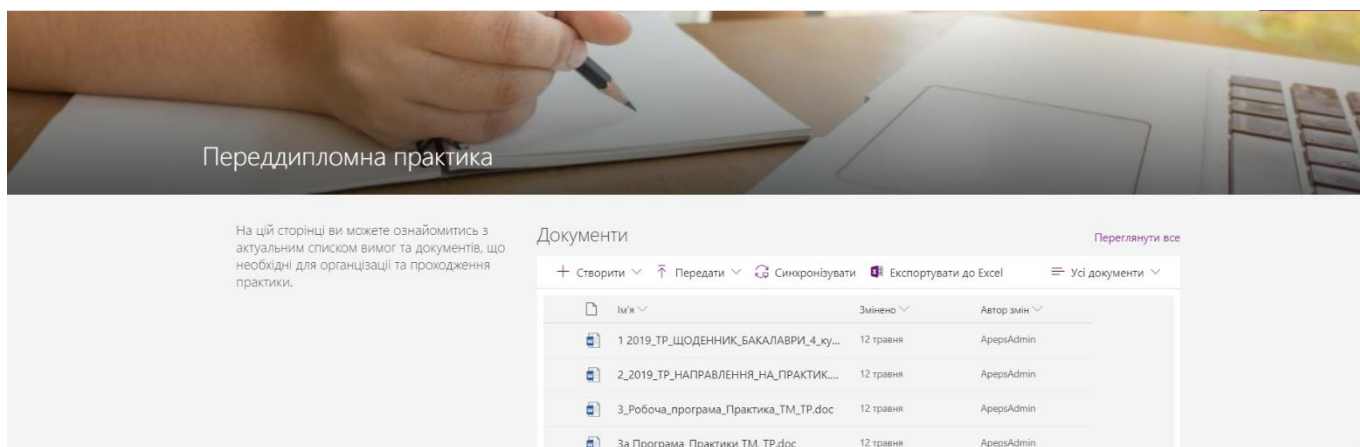


Рисунок 5.9 – Сторінка з документами для переддипломної практики

Зібравши усі матеріали в одному місці можна зекономити дуже багато часу кожному студенту і викладачу, відповідальному за процес видачі цієї інформації.

5.4. Сторінка викладача

Мета цієї сторінки – надати зручний інтерфейс для комунікації та організації роботи викладачам.

У цьому пункті буде описано функціонал роботи викладача через цю сторінку.

5.4.1. Організація своєї роботи

Викладач має доступ до особистої пошти (Рис. 5.10), за допомогою якої він може спілкуватися, в разі потреби, з іншими викладачами. Платформа Office 365 надає зручний інтерфейс пошти (Рис. 5.11). Також викладач має доступ до особистих документів (Рис. 5.12). Платформа надає зручний інтерфейс хмарного сховища (Рис. 5.13), до якого викладач має доступ з будь-якого пристрою, яке підключене до мережі Інтернет. Хмарні сховища гарантують збереження даних, що дозволяє надати їм перевагу в порівнянні з збереженням матеріалів на фізичних носіях, адже завжди є ймовірність, що фізичний носій перестане працювати, загубиться або буде несумісним з пристроєм зчитування.

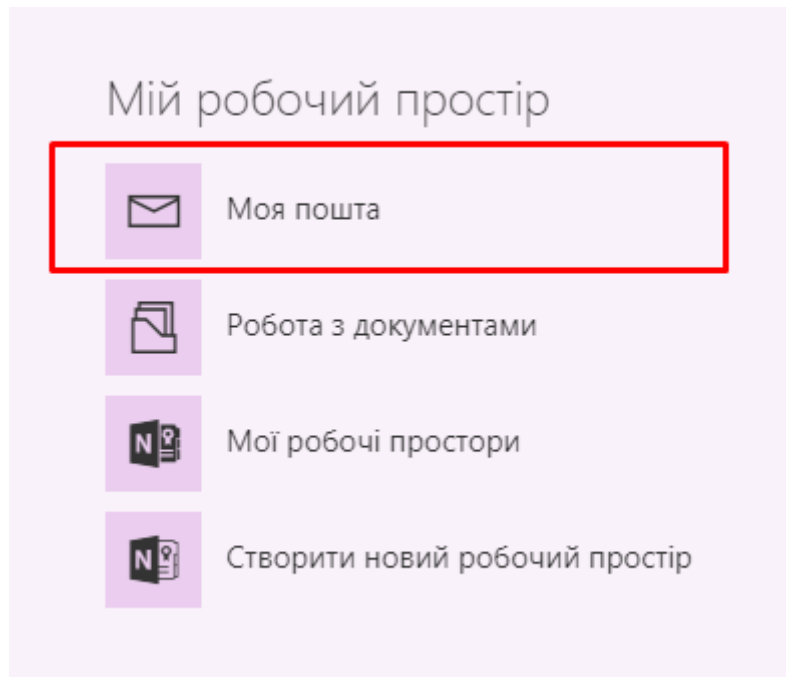


Рисунок 5.10 – Посилання на особисту поштову скриньку

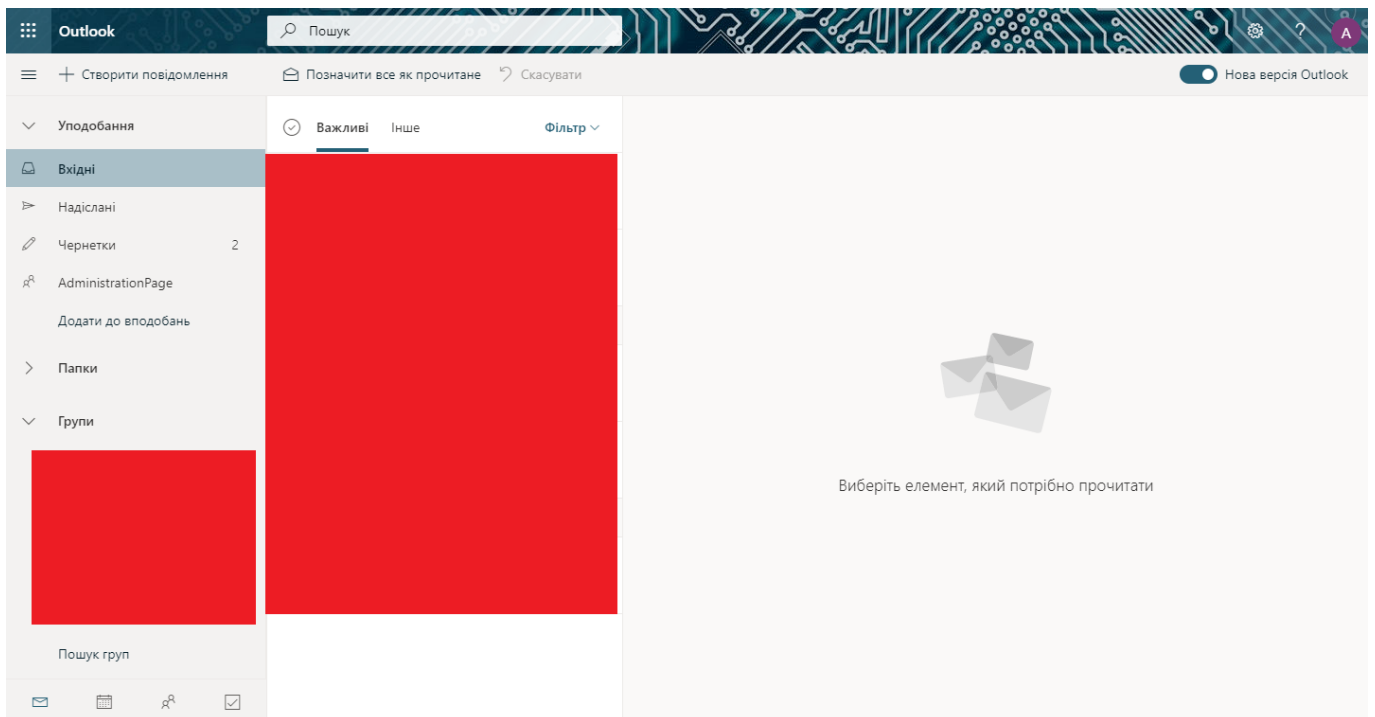


Рисунок 5.11 – Інтерфейс пошти

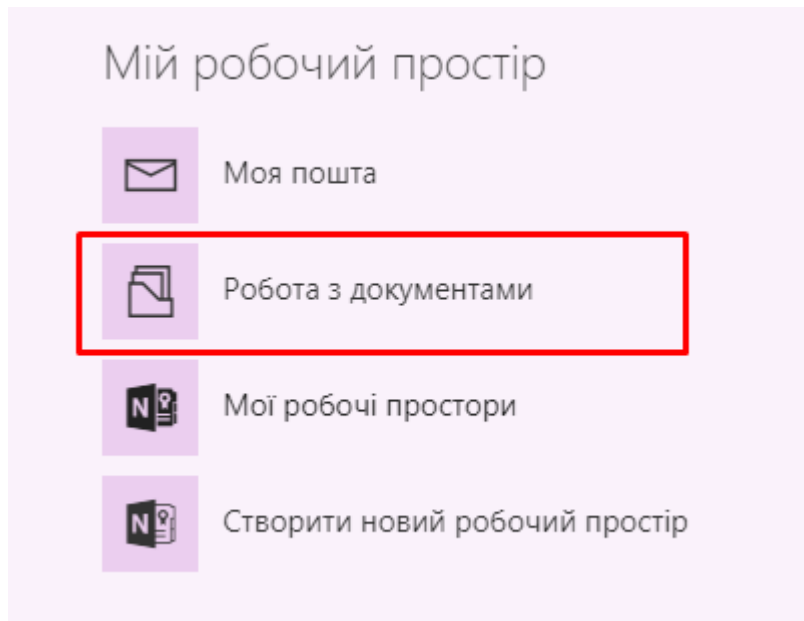


Рисунок 5.12 – Посилання на особисте хмарне сховище

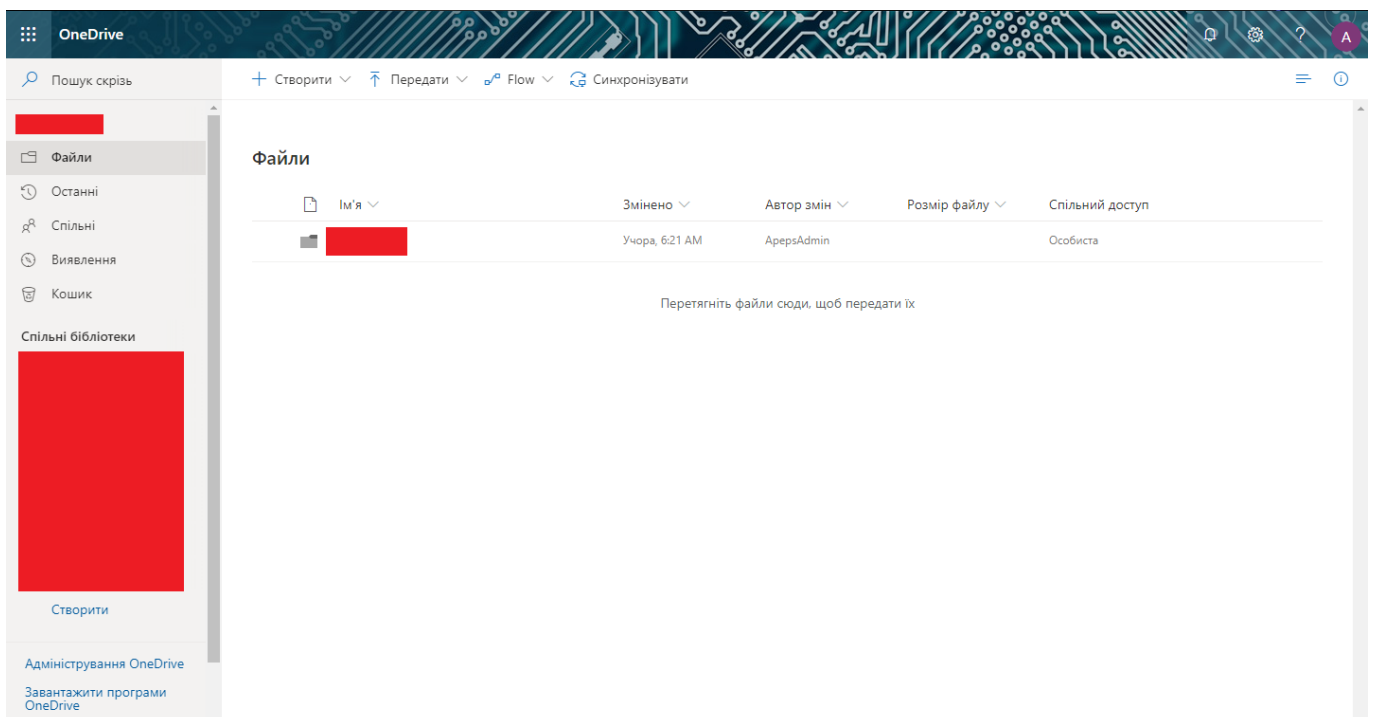


Рисунок 5.13 – Інтерфейс хмарного сховища

5.4.2. Організація роботи зі студентами

Викладач може організовувати роботу груп, яким читає предмет за допомогою робочих просторів (Рис. 5.14). Учбові середовища створюються досить швидко, просто і надають багатий функціонал для роботи з кожним окремим студентом та з групою в цілому.

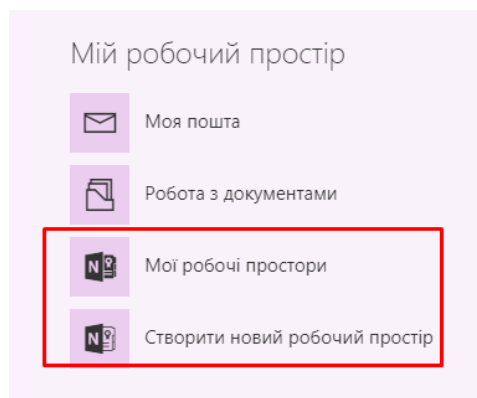


Рисунок 5.14 – Інтерфейс хмарного сховища

Для того щоб створити новий робочий простір, необхідно буде зробити декілька кроків. Інтерфейс системи буде підказувати який крок користувач робить в даний момент. Після кожного кроку треба натискати кнопку «Далі», яка буде відображатись на екрані. По-перше, треба заповнити форму (Рис. 5.15) назви простору предмету, який ви ведете у певної групи.

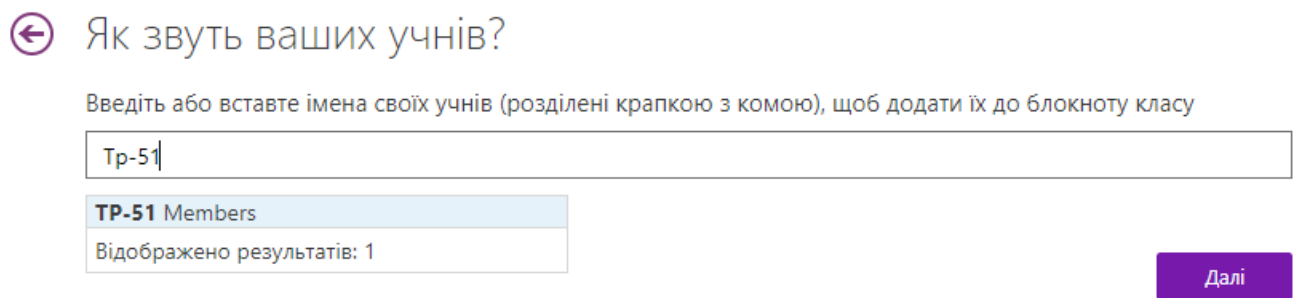
← Як називається ваш клас?

Таким буде ім'я блокнота для класу

Далі

Рисунок 5.15 – Форма назви простору предмету

Після цього вам будуть показані ваші права на цій сторінці, а також, як неонов'язковий крок, вам запропонують додати іншого викладача до робочого простору. По-друге, вам потрібно ввести назву групи (однієї або кількох) студентів, яких ви хочете додати до простору (Рис. 5.16). Механізм автодоповнення буде допомагати користувачу обрати необхідну групу. Важливо пам'ятати про те, що внесені в систему групи, за домовленістю, мають такий код, який їм було призначено деканаті.



← Як звуть ваших учнів?

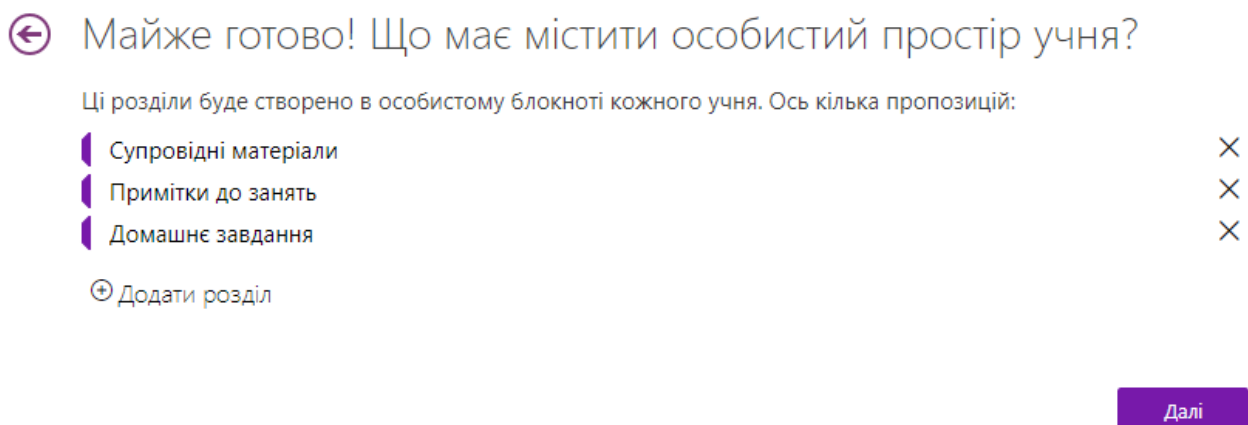
Введіть або вставте імена своїх учнів (розділені крапкою з комою), щоб додати їх до блокноту класу

ГР-51 Members
Відображено результатів: 1

Далі

Рисунок 5.16 – Форма введення груп

По-третє, ви можете налаштувати робочий простір студентів, тобто які області будуть доступні кожному студенту (Рис. 5.17).



← Майже готово! Що має містити особистий простір учня?

Ці розділи буде створено в особистому блокноті кожного учня. Ось кілька пропозицій:

- ☒ Супровідні матеріали ×
- ☒ Примітки до занять ×
- ☒ Домашнє завдання ×
- ☐ + Додати розділ

Далі

Рисунок 5.17 – Форма налаштування робочого простору студентів

На останньому кроці вам запропонують перевірити, чи все вірно і подивитися як буде виглядати простір як з сторони викладача (Рис. 5.18), так і з сторони студента (Рис. 5.19).

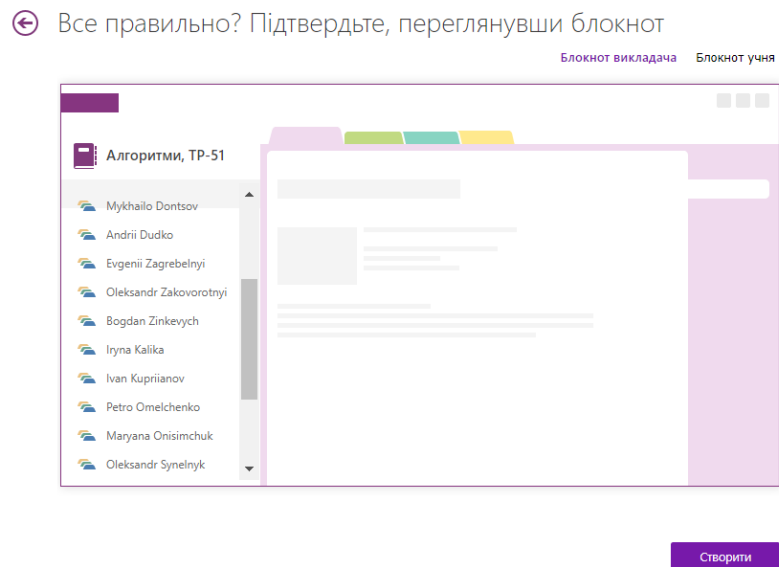


Рисунок 5.18 – Орієнтовний вигляд робочого простору зі сторони викладача

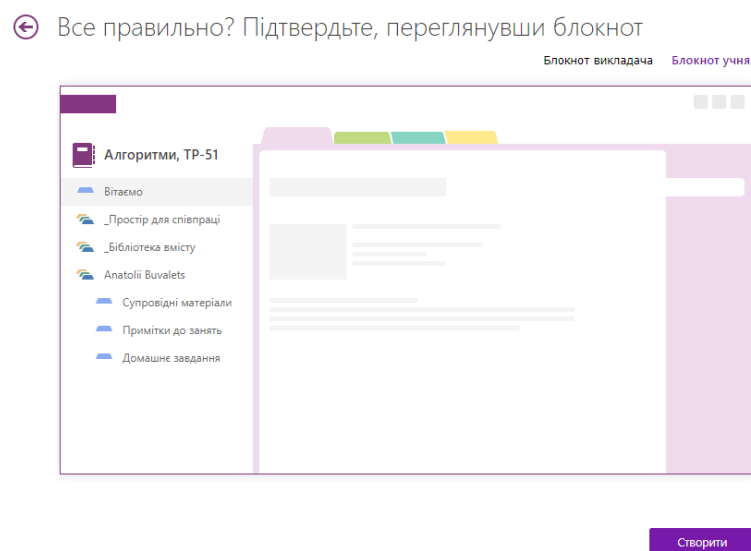


Рисунок 5.19 – Орієнтовний вигляд робочого простору зі сторони студента

На цьому етапі ви можете повернутися на будь-який попередній етап і внести зміни. Після завершення усіх налаштувань натисніть кнопку «Створити».

Тепер ми маємо можливість працювати з новоствореним робочим простором. На сторінці викладача користувач може клікнути на кнопку «Мої робочі простори» і побачити перелік своїх робочих просторів (Рис. 5.20). Для того щоб відкрити сам блокнот (робочий простір), клікніть на «Відкрити у OneNote Online», щоб побачити його інтерфейс (Рис. 5.21). Вам буде запропоновано переглянути коротке відео для ознайомлення з системою (у верхньому правому кутку).

Ваші блокноти



Алгоритми, ТР-51

[Відкрити у OneNote Online](#)

Рисунок 5.20 – Список блокнотів (робочих просторів) викладача

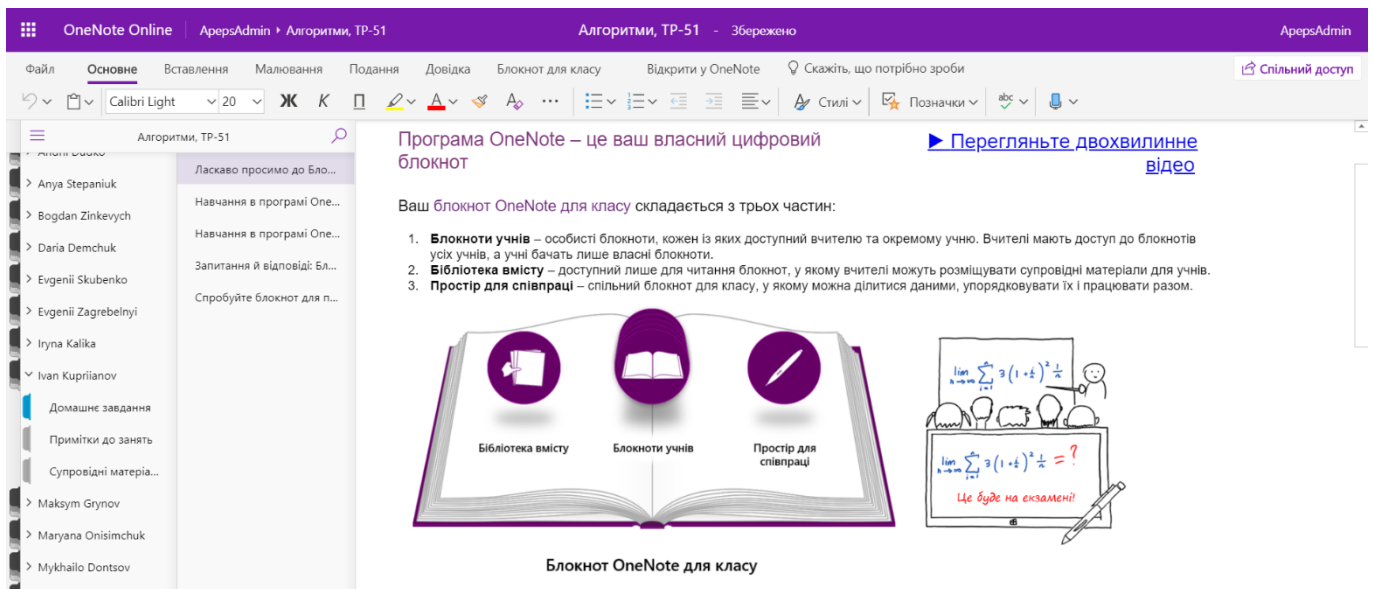


Рисунок 5.21 – Інтерфейс робочого простору з сторони викладача

Для покращення навичок роботи з системою блокнотів OneNote та подальшого ознайомлення з системою можна знайти багато ресурсів в мережі Інтернет.

Коли студенти вже є частиною робочого простору, їм достатньо зайти в систему Microsoft Office 365 і перейти до OneNote Online. Вони будуть мати це посилання у списку їх робочих просторів на вкладці «Мені надано доступ» (Рис. 5.22).

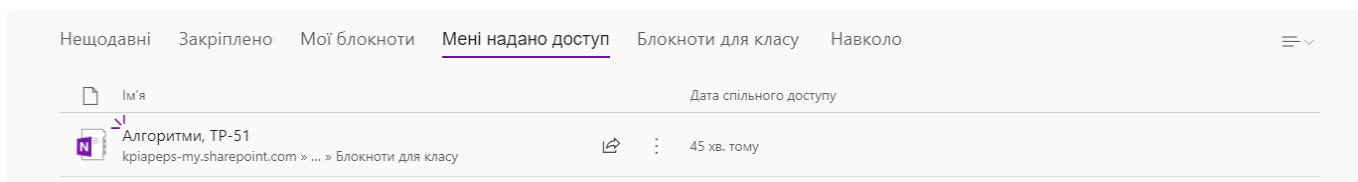


Рисунок 5.22 – Вкладка «Мені надано доступ» на OneNote Online

Якщо перейти в цей блокнот, то можна почати працювати з робочим простором як студент (Рис. 5.23).

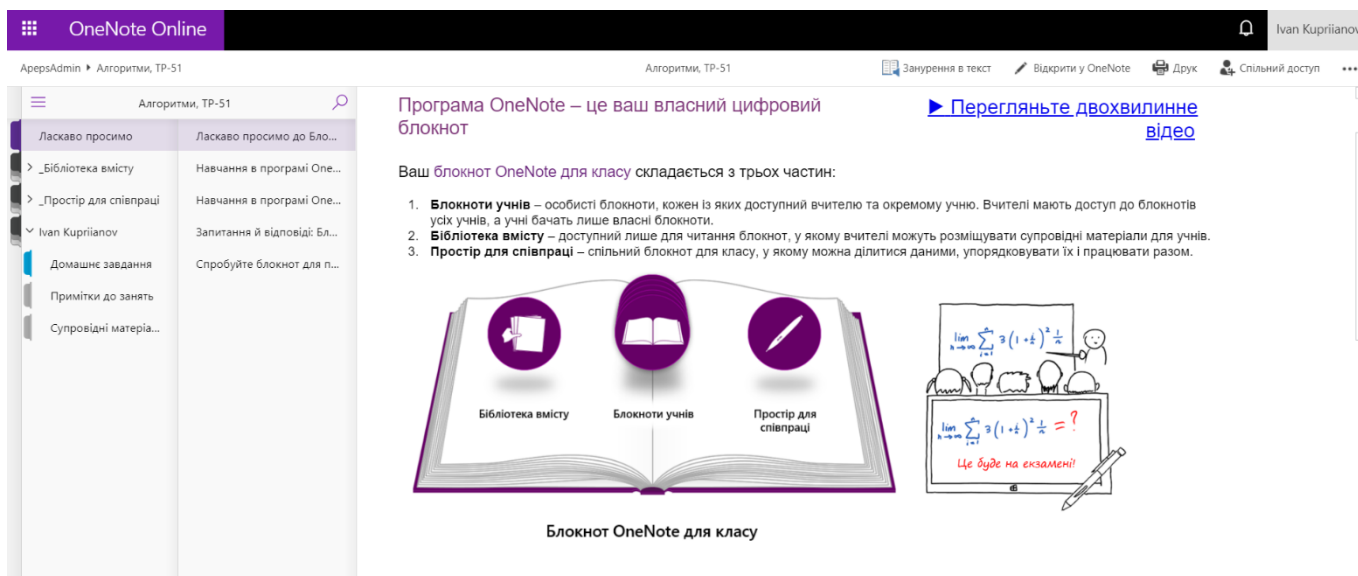


Рисунок 5.23 – Інтерфейс робочого простору з сторони студента

Аналогічно, як і викладач в студента є посилання на відео-ознайомлення з системою OneNote Online.

5.4.3. Організовувати роботу групи викладачів

Організація роботи декількох викладачів буває просто необхідною. Може бути необхідність організувати зустріч таким чином, щоб її бачили усі викладачі, або лише певні люди. Причому, потрібно не просто організувати зустріч, а й повідомити кожного викладача. Для того щоб уникнути незручностей можна використовувати вбудовані в Office 365 властивості календаря (Рис. 5.24).

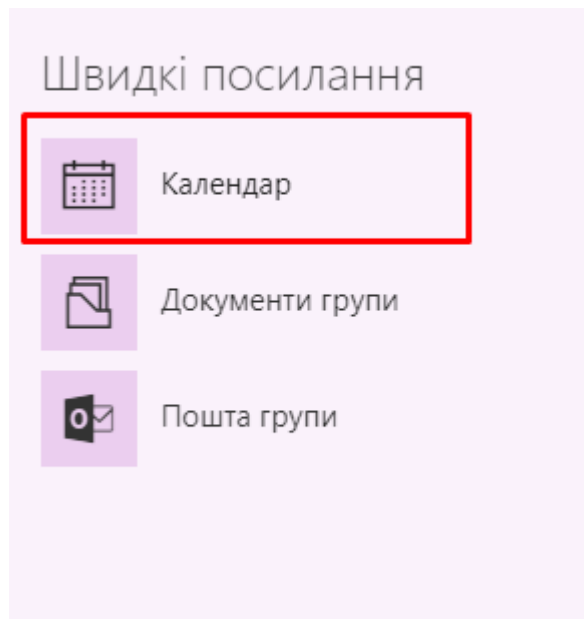


Рисунок 5.24 – Посилання на календар групи

Можна організувати зустріч (запланувати щось для себе або для групи викладачів), вона буде відображена в календарі (Рис. 5.25) і на сторінці викладачів.

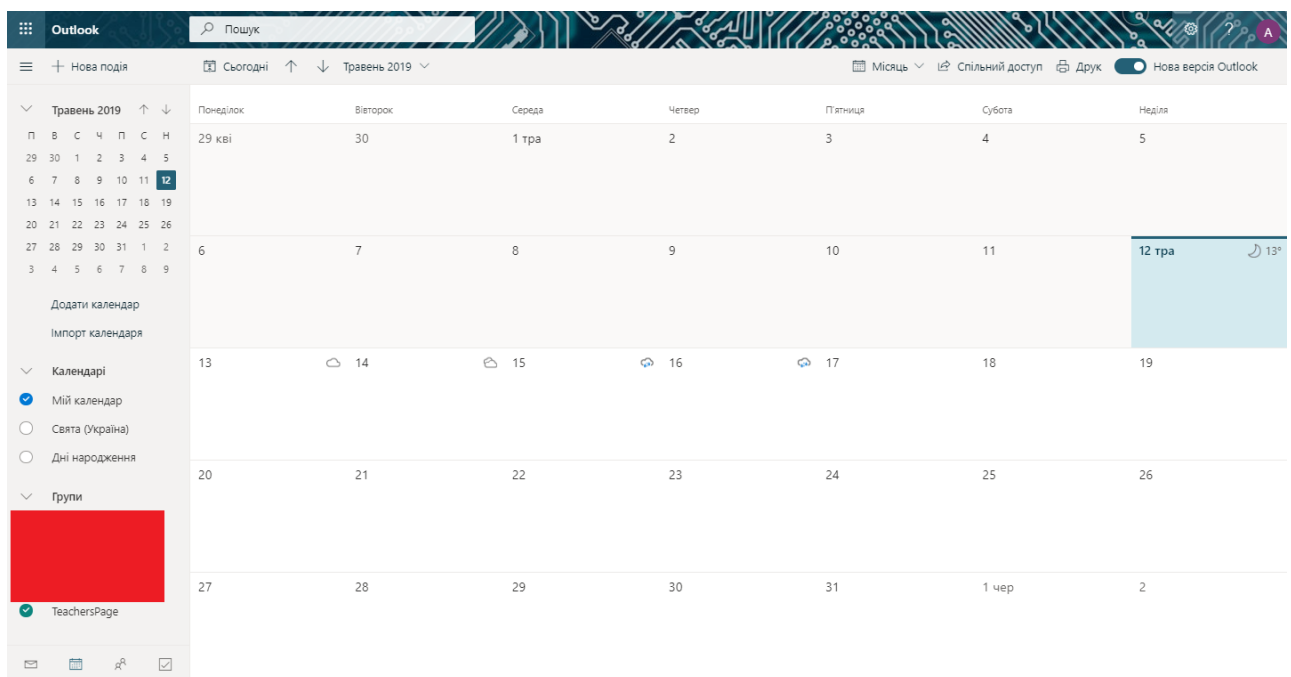


Рисунок 5.25 – Інтерфейс календаря групи

Для того щоб створити подію, потрібно обрати дату в календарі та клікнути. Відкриється діалогове вікно (Рис. 5.26).

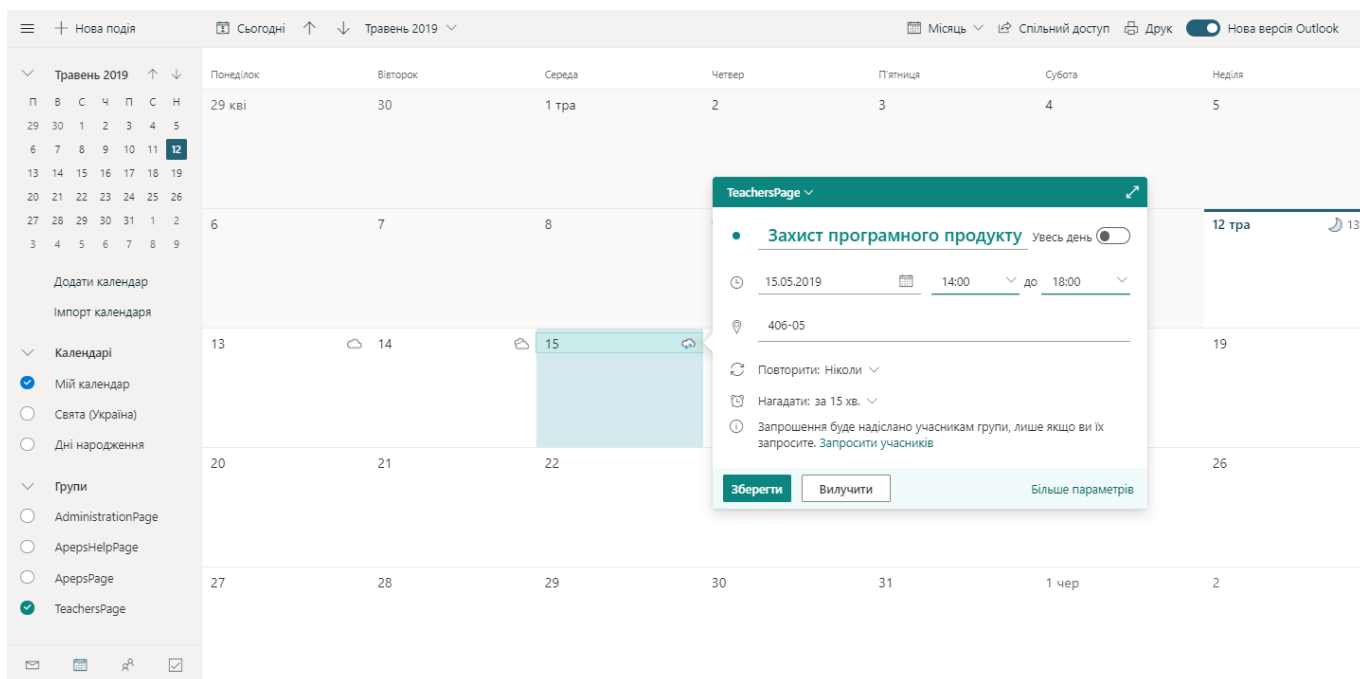


Рисунок 5.26 – Діалогове вікно створення події

Далі користувачу потрібно ввести інформацію про подію. В верхньому лівому кутку діалогового вікна можна обрати для кого організувати подію – для себе («Мій календар») або для вчителів («TeachersPage»).

Наразі можна зберегти подію (або скасувати її). Також можна натиснути кнопку «Запросити учасників» або «Більше параметрів». В цих випадках перед користувачем з'явиться наступне діалогове вікно (Рис. 5.28). Можна додати інших учасників в подію, уточнити деталі події. Після цього потрібно натиснути кнопку «Надіслати».

Тепер подія буде відображатися в календарі користувача і на сторінці TeachersPage, якщо вона була створена для групи викладачів (Рис 5.27).

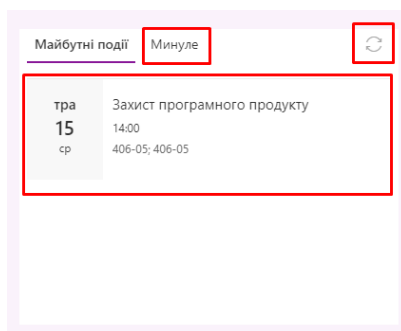


Рисунок 5.27 – Подія на сторінці викладачів

На сторінці викладачів можна переглянути майбутні та минулі події групи. Якщо подія тільки що була організована, а вона ще не відобразилася, потрібно почекати і оновити веб-частину (кнопка в правому верхньому кутку веб-частини на сторінці). Також можна переглянути подробиці події (Рис. 5.30), клікнувши на неї (Рис. 5.27).

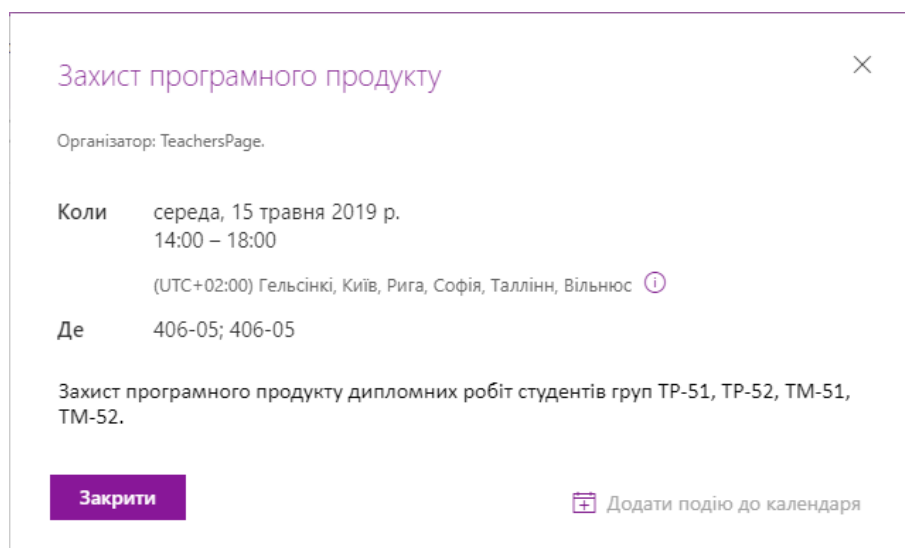


Рисунок 5.30 – Діалогове вікно з подробицями події

Також сторінка викладачів має доступ до документів групи (Рис. 5.31) та пошти групи (Рис. 5.32). Відмінність документів групи від особистих документів в тому, що вони спільні для всіх користувачів, що належать до групи викладачів.

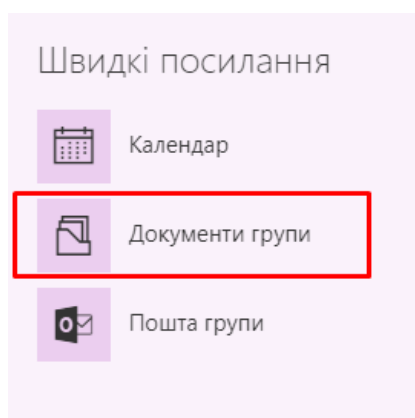


Рисунок 5.31 – Посилання на документи групи

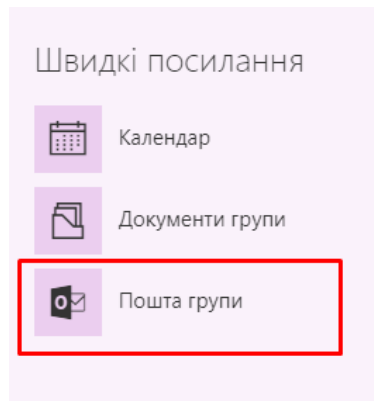


Рисунок 5.32 – Посилання на пошту групи

Пошта вашої групи дозволяє вам організовувати обмін листами в середині групи і від імені групи. На відміну від особистої пошти, ви бачите, що вибраний пункт меню не «Вхідні», а «TeachersPage» (Рис 5.33).

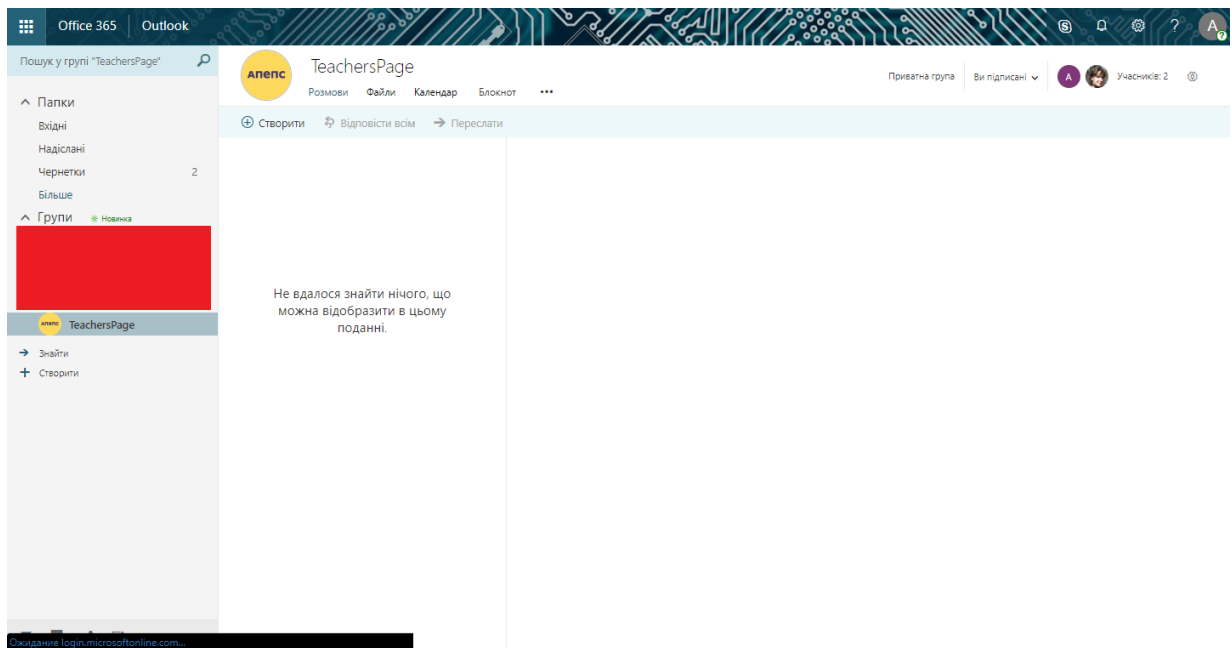


Рисунок 5.33 – Інтерфейс пошти групи

Можна відкрити пошту групи не тільки через конкретне посилання. Досягти такого ж ефекту можна, якщо перейти на особисту пошту, а потім обрати необхідну групу.

5.4.4. Інтерфейс адміністратора

Інтерфейс має досить вузький функціонал, який націлений на керування користувачами. В цілому, інтерфейс цієї сторінки можна поділити на 3 частини (рис. 5.34):

- документи;
- особисті посилання;
- швидкі посилання для роботи.

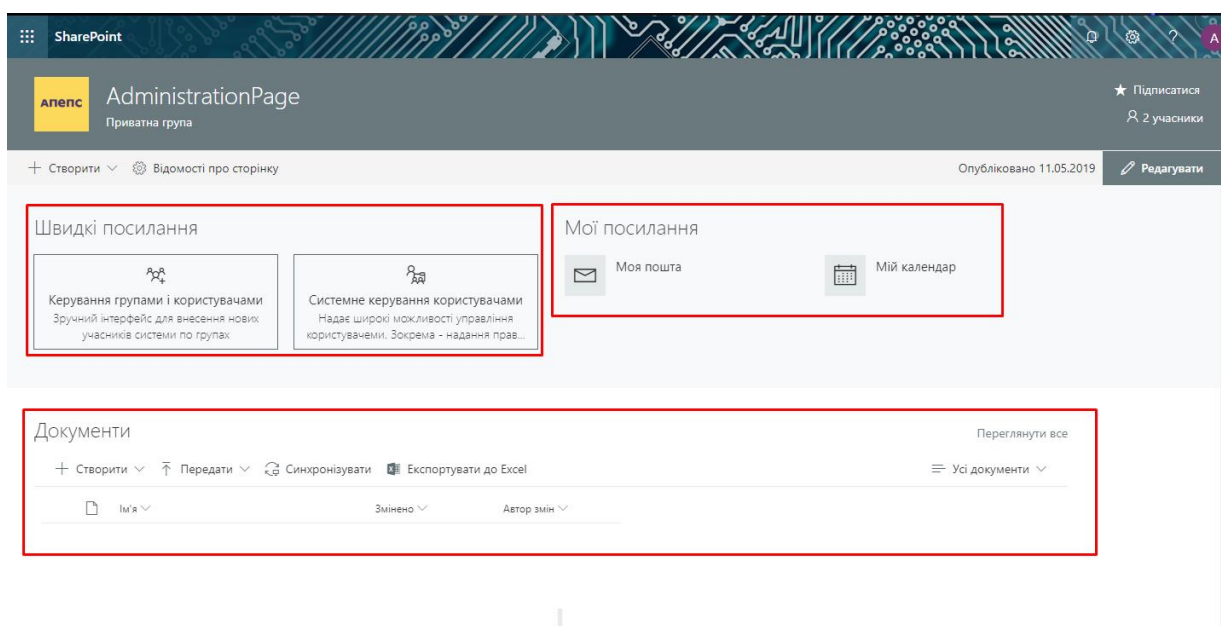


Рисунок 5.34 – Інтерфейс сторінки адміністратора

Посилання «Мій календар» і «Моя пошта» ведуть відповідно на особистий календар і особисту пошту користувача з правами адміністратора. Веб-частина «Документи» надає доступ до документів сторінки адміністратора.

Швидкі посилання надають доступ до функціоналу керування користувачами. Посилання «Керування групами і користувачами» веде до інтерфейсу за допомогою якого можна зручно вносити нових користувачів в систему по групах, надавати їм ліцензії і т.е. (Рис. 5.35). Цей функціонал було розроблено для роботи зі специфікою кафедр роботи вузів, оскільки вбудований функціонал Microsoft Office 365 не підходить для цих цілей.

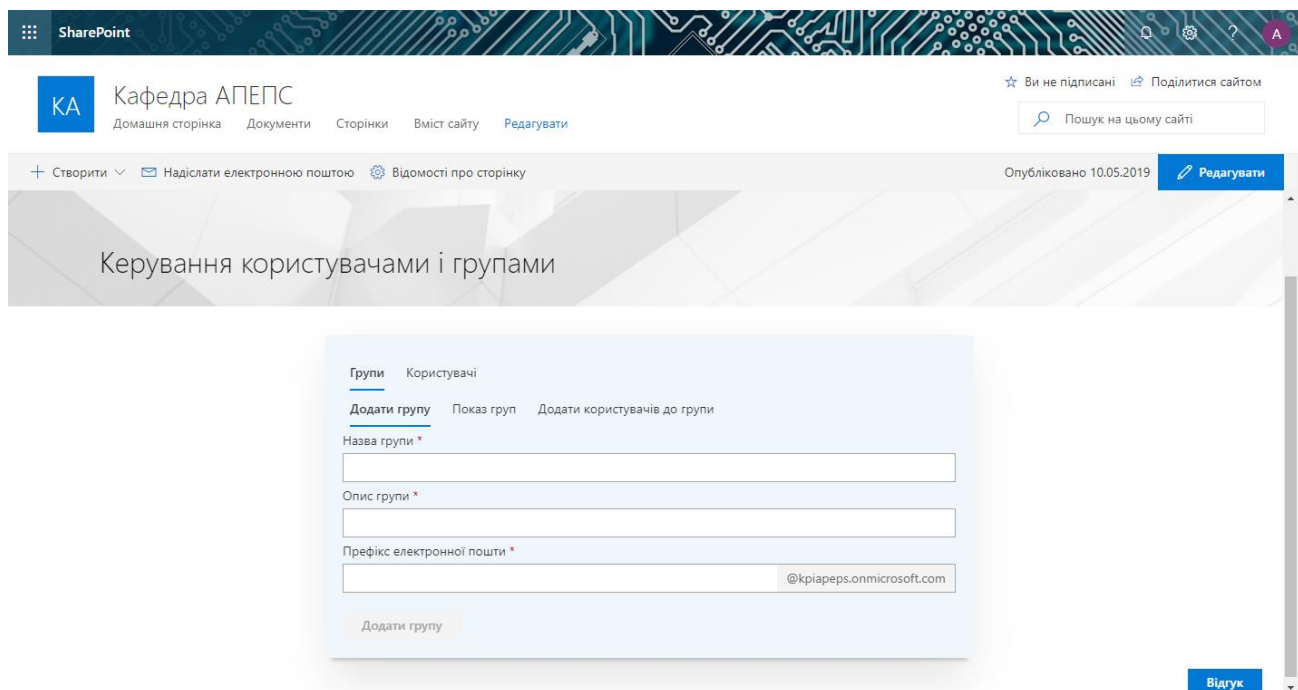


Рисунок 5.35 – Інтерфейс сторінки адміністратора

Швидке посилання «Системне керування користувачами» веде до інтерфейсу адміністратора Microsoft Office 365 (Рис. 5.36). Цей інтерфейс надає повний і детальний доступ до даних кожного користувача, пошук, фільтрацію по користувачах, їх експорт. Цей інтерфейс також дозволяє імпортувати користувачів в стандартному форматі. Приведено деякі важливі функції: скинути пароль користувача, призначити користувача групі, змінити ім'я користувача, змінити електронну пошту користувача, налаштувати ліцензії користувача, примусовий вихід користувача з системи, блокування користувача, відновлення користувача (Рис. 5.37).

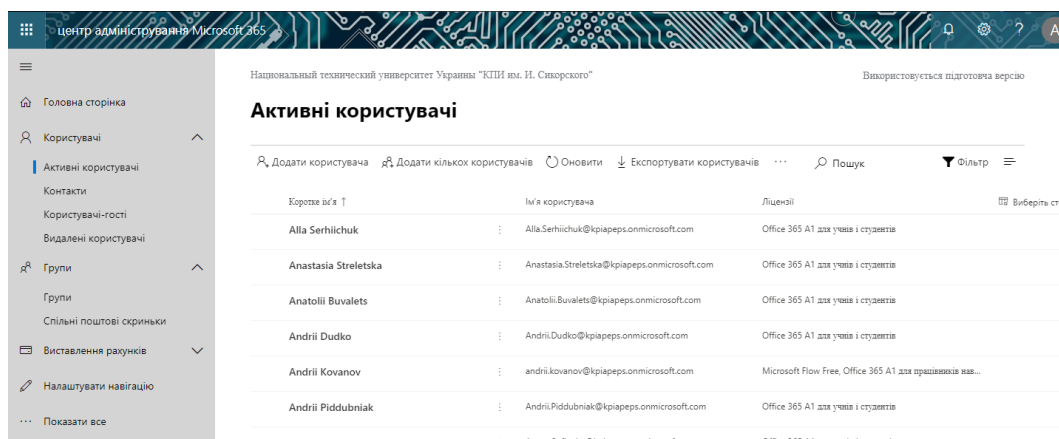


Рисунок 5.36 – Інтерфейс системної сторінки адміністратора

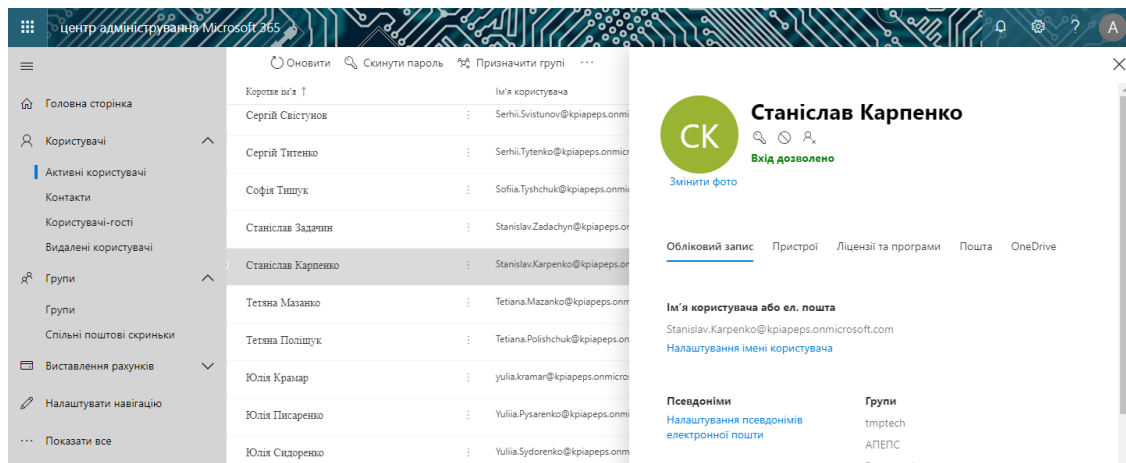


Рисунок 5.37 – Інтерфейс функцій для конкретного користувача

5.5. Вікі-сторінка

Інтерфейс вікі-сторінки створений як база знань для розробленого додатку. Він містить кілька дочірніх сторінок, згідно структури додатку (Рис. 5.38). Припускається, що ця сторінка буде завжди оновлена згідно поточного стану

Ласкаво просимо до вікі!

- Головна сторінка кафедри
- Сторінка адміністратора
- Сторінка викладачів
- Простір студентів

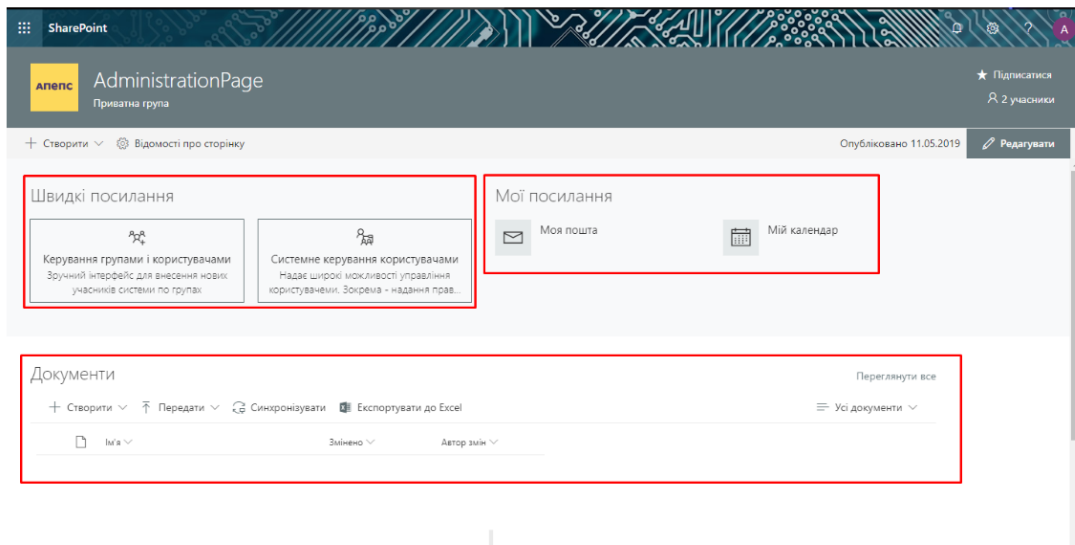
Рисунок 5.38 – Інтерфейс вікі-сторінки

Кожна з сторінок описує відповідну сторінку додатку. Так, посилання «Головна сторінка кафедри» веде на вікі-сторінку головної сторінки кафедри АПЕПС (Рис. 5.39).

Припускається, що ці сторінки (як частина додатку) і в подальшому будуть відповідати структурі додатку, тому що вони є допомогою новим користувачам в освоєнні в новій для них системі.

Сторінка AdministrationPage

На цій сторінці ви можете управляти не тільки своєю поштою, документами і органайзером, а і перейти на ресурси управління іншими користувачами.



• Керування групами і користувачами

На даній сторінці ви можете керувати групами і користувачами системи. На даній сторінці ви можете керувати групами і користувачами системи.

Рисунок 5.39 – Інтерфейс вікі-сторінки адміністратора

Ця сторінка відкрита для редагування тільки користувачам з правами адміністратора.

5.6. Висновки до розділу

У даному розділі було детально розглянуто користувацький інтерфейс системи та описано можливості та функції, які надає цей інтерфейс.

ВИСНОВКИ

Системи інформаційного учбового середовища стають все більш затребуваними в процесі навчання на сьогоднішній день. У сучасному світі дистанційна форма навчання має ряд переваг над традиційною. Для того щоб наблизити цей процес в українських вузах було запропоновано і розроблено модель системи інформаційного учбового середовища.

— В процесі розробки додатку було проведено наступні роботи:

— Проаналізовано існуючі системи дистанційного навчання Moodle і ODEMS;

— Розглянуто технології створення додатків на базі платформи Microsoft Office 365;

— Проаналізовано бажаний результат розробки додатку, виведено критерії;

— Проаналізовано сучасні технології розробки і обрано стек технологій, за допомогою якого в подальшому розроблявся додаток;

— Описано функціональні вимоги до розроблюваного додатку;

— Проаналізовано методології і практики програмування, які використовуються на підприємстві, на якому проходила практика;

— Розроблено програмний продукт.

Кінцевий продукт має наступний функціонал:

Інтерфейс адміністратора, що надає можливості повного керування в середині розробленої інфраструктури;

Інтерфейс викладача, що надає функціонал власного робочого кабінету і функціонал взаємодії зі студентами. Останній є кроком до дистанційного навчання

Інтерфейс студента, який носить інформативний характер і надає можливості адміністраторам системи надавати контент, корисний студентам в одному місці

Інтерфейс вікі-сторінок, розроблений як база знань і призначений здебільшого для нових користувачів в системі.

В додатку наявні надійна система авторизації та розподіл по ролях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація Moodle [Електронний ресурс] – Режим доступу до ресурсу: <https://moodle.dduvs.in.ua/MoodleDocs.php>.
2. Moodle Plugins [Електронний ресурс] – Режим доступу до ресурсу: <https://moodle.org/plugins/>.
3. Компанія Microsoft випустила для Linux середовище розробки Visual Studio Code [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <http://www.opennet.ru/opennews/art.shtml?num=42132>.
4. SharePoint Collaboration Tools [Електронний ресурс] – Режим доступу до ресурсу: <https://products.office.com/en-us/sharepoint/collaboration>.
5. SharePoint Plans [Електронний ресурс] – Режим доступу до ресурсу: <https://products.office.com/en-us/SharePoint/compare-sharepoint-plans?legRedir=true&CorrelationId=61eeac9c-61f4-4f47-8f0e-e273392f77b8>.
6. SharePoint Framework Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/sharepoint-framework-overview>.
7. JavaScript: The Definitive Guide / D. Flanagan — O'Reilly. 2011 — 13 с.
8. Usage of JavaScript for websites [Електронний ресурс] – Режим доступу до ресурсу: <https://w3techs.com/technologies/details/cp-javascript/all/all>.
9. React: Components and Props [Електронний ресурс] – Режим доступу до ресурсу: <https://reactjs.org/docs/components-and-props.html>.
10. Refs and DOM [Електронний ресурс] – Режим доступу до ресурсу: <https://reactjs.org/docs/refs-and-the-dom.html>.
11. Microsoft Graph Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.microsoft.com/en-us/graph/docs/concepts/overview>.
12. Підключення додатків на базі SharePoint Framework до ресурсів захищених Azure AD [Електронний ресурс] – Режим доступу до ресурсу:

<https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/guidance/connect-to-api-secured-with-aad>.

13. Azure AD для автентифікації в SharePoint [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/office365/enterprise/using-azure-ad-for-sharepoint-server-authentication>.

ДОДАТОК А

Інформаційне середовище кафедри АПЕПС на базі Office 365

Специфікація

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ТР5161_19Б

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТР5161_19Б 81-1	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТР5161_19Б 12-1	group-user-manager-reworked.rar	Основний компонент
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТР5161_19Б 12-2	admin.rar	Дані про обліковий запис адміністратора
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТР5161_19Б 13-1	Довідка.doc	Довідка про програму

ДОДАТОК Б

Інформаційне середовище кафедри АПЕПС на базі Office 365

Текст програми

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТР5161_19Б 12-1

Аркушів 7

Київ 2019

```

import * as React from "react";
import styles from "../UpcomingMeetings.module.scss";
import { IGroupsUserManagerProps } from "../IGroupsUserManagerProps";
import { escape } from "@microsoft/sp-lodash-subset";
import {
  /* GraphHttpClient,
  IGraphHttpClientOptions,*/
  HttpClientResponse,
  SPHttpClient
} from "@microsoft/sp-http";
import {
  BaseClientSideWebPart,
  IWebPartContext,
  PropertyPaneSlider
} from "@microsoft/sp-webpart-base";
import { GroupService } from "../services/GroupService";
import AddGroup from "../AddGroup/AddGroup";
import { Pivot, PivotItem } from "office-ui-fabric-react/lib/Pivot";
import { Label } from "office-ui-fabric-react/lib/Label";
import { DetailsList } from "office-ui-fabric-react/lib/DetailsList";
import AddUser from "../AddUser/AddUser";
import { IGroupsUserManagerState } from "../IGroupsUserManagerState";
import { AuthService } from "../services/AuthService";
import * as AuthenticationContext from 'adal-angular';
import adalConfig from '../AdalConfig';
import { IAdalConfig } from '../../IAdalConfig';
import '../../WebPartAuthenticationContext';
import { GroupList } from "../GroupList/GroupList";
import { UserList } from "../UserList/UserList";
import { GroupUserList } from "../GroupUserList/GroupUserList";
import { Resource } from "../Util/Resource";
import Modal from "office-ui-fabric-react/lib/Modal";

export class AuthService {
  private _authorizationContext: adal.AuthenticationContext;

  constructor(authContext: adal.AuthenticationContext) {
    this._authorizationContext = authContext;
  }

  public getMSGraphAccessToken(): Promise<string> {
    return new Promise<string>((resolve: (accessToken: string) => void, reject:
(error: any) => void): void => {
      const graphResource: string = 'https://graph.microsoft.com';
      const accessToken: string =
this._authorizationContext.getCachedToken(graphResource);
      console.info("Token retrieve started!");
      if (accessToken) {
        console.info("Found cached token: ");
        resolve(accessToken);
        return;
      }

      if (this._authorizationContext.loginInProgress()) {
        let msg = 'Login already in progress';

```

```

        console.log(msg)
        reject(msg);
        return;
    }
    this._authAuthorizationContext.acquireToken(graphResource, (error: string,
token: string) => {
        console.info("Acquiring token for Graph from authContext");
        if (error) {
            console.info(error);
            reject(error);
            return;
        }

        if (token) {
            console.info("Retrieved new token: " + token);
            resolve(token);
        }
        else {
            console.info('Couldn\'t retrieve access token');
            reject('Couldn\'t retrieve access token');
        }
    });
});
}
}

```

```

export default class GlobalModal extends React.Component<IGlobalModalProps,
IGlobalModalState> {
    private static Modal: GlobalModal;

    public constructor(props) {
        super(props);

        this.state = {
            showModal: false,
            showLoading: false,
            text: ""
        };
        GlobalModal.Modal = this;
        console.log(this);
        console.log(GlobalModal);
    }

    public render(): React.ReactElement<{}> {
        return <Modal
            isOpen={this.state.showModal}
            onDismiss={this._closeModal.bind(this)}
            isBlocking={false}>
            <div style={{ padding: "20px" }}>
                <p style={{display: this.state.showLoading ? "block" : "none"}}>
                    <ProgressIndicator
                        label='Loading'
                    />
                </p>
                <p>{this.state.text}</p>
            </div>
        </Modal>
    }
}

```

```

                <p style={{display: !this.state.showLoading ? "block" : "none"}}>
                    <Button                                text={Resource.OkLabel}
onClick={this._closeModal.bind(this)} />
                </p>
            </div>
        </Modal>;
    }

    public static show(text: string) {
        GlobalModal.Modal.setState(prevState => {
            prevState.showModal = true;
            prevState.showLoading = false;
            prevState.text = text;
            return prevState;
        });
    }

    public static showLoading(): void {
        GlobalModal.Modal.setState(prevState => {
            prevState.showModal = true;
            prevState.showLoading = true;
            return prevState;
        });
    }

    private _closeModal() {
        this.setState(prevState => {
            prevState.showModal = false;
            prevState.showLoading = false;
            prevState.text = "";
            return prevState;
        });
    }
}

export default class GroupsUserManager extends
React.Component<IGroupsUserManagerProps, IGroupsUserManagerState> {

    private _groupService: GroupService;
    private authenticationContext: adal.AuthenticationContext;

    public constructor(props) {
        super(props);
        console.log(this.props);

        this.state = {
            loading: false,
            error: null,
            signedIn: false,
            showInfoModal: false
        };

        const config: IAdalConfig = adalConfig;
        config.popUp = true;
        config.webPartId = this.props.webPartId;
        config.callback = (error: any, token: string): void => {

```

```

        this.setState((previousState) => {
            previousState.error = error;
            previousState.signIn
        });
    });
    (!this.authenticationContext.getCachedUser());
    return previousState;
    });
};

this.authenticationContext = new AuthenticationContext(config);
AuthenticationContext.prototype._singletonInstance = undefined;
}

public componentDidMount(): void {
    this.authenticationContext.handleWindowCallback();

    if (window !== window.top) {
        return;
    }

    this.setState((previousState) => {
        previousState.error = this.authenticationContext.getLoginError();
        previousState.signIn = (!this.authenticationContext.getCachedUser());
        return previousState;
    });
}

public signIn(): void {
    this.authenticationContext.login();
}

private _getLoginButton() {
    let signInSpan = <span className={styles.buttonLabel}>Sign in</span>;
    let signInHint = <span className={styles.buttonDescription}>Sign in to
interact with this app</span>;
    return <button className={` ${styles.button} ${styles.buttonCompound}`}
        onClick={() => { this.signIn(); }}
        style={{ marginLeft: "auto", marginRight: "auto" }}>
        {signInSpan}
        {signInHint}
    </button>;
}

public render(): React.ReactElement<IGroupsUserManagerProps> {
    const login: JSX.Element = this.state.signIn
        ? <div />
        : this._getLoginButton()
    const loading: JSX.Element = this.state.loading ? <div style={{ margin: '0
auto', width: '7em' }}><div className={styles.spinner}><div
className={` ${styles.spinnerCircle} ${styles.spinnerNormal}`}></div><div
className={styles.spinnerLabel}>Loading...</div></div></div> : <div />;
    const error: JSX.Element = this.state.error ? <div><strong>Error: </strong>
{this.state.error}</div> : <div />;

    let body = <div />;

```

```

    if (this.state.signedIn &&
        !this.state.loading &&
        !this.state.error) {
        body = <div className={styles.container} >
            <div className={styles.row}>
                <Pivot>
                    <PivotItem linkText={Resource.GroupsLabel}>
                        <Pivot>
                            <PivotItem linkText={Resource.AddGroupLabel}>
                                <AddGroup      httpClient={this.props.httpClient}
authContext={this.authenticationContext} />
                            </PivotItem>
                            <PivotItem linkText={Resource.ShowGroupLabel}>
                                <GroupList      httpClient={this.props.httpClient}
authContext={this.authenticationContext} />
                            </PivotItem>
                            <PivotItem linkText={Resource.AddUsersGroupLabel}>
                                <AddUsersGroup
httpClient={this.props.httpClient} authContext={this.authenticationContext} />
                            </PivotItem>
                        </Pivot>
                    </PivotItem>
                    <PivotItem linkText={Resource.UsersLabel}>
                        <Pivot>
                            <PivotItem linkText={Resource.AddUserLabel}>
                                <AddUser
webPartContext={this.props.webPartContext}      httpClient={this.props.httpClient}
webPartId={this.props.webPartId} />
                            </PivotItem>
                            <PivotItem linkText={Resource.ShowUsersLabel}>
                                <UserList      httpClient={this.props.httpClient}
authContext={this.authenticationContext} />
                            </PivotItem>
                            <PivotItem linkText={Resource.ShowGroupsWithUsers}>
                                <GroupUserList
httpClient={this.props.httpClient} authContext={this.authenticationContext} />
                            </PivotItem>
                        </Pivot>
                    </PivotItem>
                </Pivot>
            </div>
            <GlobalModal />
        </div>;
    }

    return (
        <div className={styles.upcomingMeetings}>
            {login}
            {loading}
            {error}
            {body}
        </div>
    );
}

```

```

private _closeModal(): void {
    this.setState(prevState => {
        prevState.showInfoModal = false;
        return prevState;
    })
}

private getMSGraphAccessToken(): Promise<string> {
    return new Promise<string>((resolve: (accessToken: string) => void, reject:
(error: any) => void): void => {
        const graphResource: string = 'https://graph.microsoft.com';
        const accessToken: string =
this.authenticationContext.getCachedToken(graphResource);
        if (accessToken) {
            resolve(accessToken);
            return;
        }

        if (this.authenticationContext.loginInProgress()) {
            reject('Login already in progress');
            return;
        }

        this.authenticationContext.acquireToken(graphResource, (error: string,
token: string) => {
            if (error) {
                reject(error);
                return;
            }

            if (token) {
                resolve(token);
            }
            else {
                reject('Couldn\'t retrieve access token');
            }
        }));
    });
}

```


ДОДАТОК В

Інформаційне середовище кафедри АПЕПС на базі Office 365

Адміністративні дані

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ТР5161_19Б 12-2

Аркушів 2

Київ 2019

ДОДАТОК Г

Інформаційне середовище кафедри АПЕПС на базі Office 365

Опис програми

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТР5161_19Б 13-1

Аркушів 10

Київ 2019

АНОТАЦІЯ

Додаток містить опис одного з компонентів системи - компонент інтерфейсу адміністратора для управління обліковими записами, що виконує деякі із завдань, поставлених в розділі 1, а саме:

- Забезпечення обмеження доступу до функціоналу додатку
- Імпорт груп користувачів та створення облікових записів із зовнішніх даних, а саме з списку в форматі Excel;
- Реалізація зручного інтерфейсу для доступу до функціоналу додатку.

Додаток розроблений за допомогою мови програмування TypeScript та фреймворку React у середовищі програмування Microsoft Visual Studio Code.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ	78
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ.....	79
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ	80
4. ТЕХНІЧНІ ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ	81
5. ВИКЛИК І ЗАВАНТАЖЕННЯ.....	82
6. ВХІДНІ ДАНІ	83
7. ВИХІДНІ ДАНІ.....	84

ЗАГАЛЬНІ ВІДОМОСТІ

У цьому додатку міститься опис основного компоненту системи управління обліковими записами, що виконує деякі із завдань, поставлених в розділі 1. У додатку Б міститься лістинг програмного коду компоненту.

Система працює в середовищі Office 365 SharePoint Online, та для роботи потребує браузер, що підтримує JavaScript та Cookies.

Додаток розроблений за допомогою мови програмування TypeScript та фреймворку React у середовищі програмування Microsoft Visual Studio Code.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Додаток надає функціонал для роботи з індивідуальними обліковими записами та групами користувачів, а саме:

- створення, редагування та видалення індивідуальних облікових записів для студентів та викладачів;
- створення груп за напрямленнями;
- додавання користувачів до груп;
- імпорт груп користувачів із зовнішнього середовища за допомогою списків в форматі Excel;
- перегляд усіх облікових записів та груп, пошук облікових записів;
- експорт облікових записів в форматі Excel.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Згідно з архітектурою, додаток складається з трьох головних модулів:

- шар сервісів;
- шар моделей;
- шар компонентів.

Шар сервісів містить в собі декілька класів, що слугують для виклику зовнішніх служб Microsoft Graph, Azure AD та роботи з списками користувачів в форматі Excel.

Шар моделей містить в собі класи та інтерфейси, що використовуються для моделювання сутностей, якими оперує додаток.

Шар компонентів містить компоненти користувацького інтерфейсу, реалізовані за допомогою фреймворку React. Вони використовують класи шару сервісів, щоб оперувати даними про облікові записи користувачів та груп.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для використання додатку користувач повинен мати обліковий запис в електронному середовищі кафедри Office 365 з правами адміністратора, персональний комп'ютер або мобільний пристрій з браузером, що підтримує JavaScript, а також підключення до мережі Інтернет.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Програма не потребує інсталяції і доступна на сайті електронного інформаційного середовища кафедри.

ВХІДНІ ДАНІ

Вхідна інформація для додатку:

- а) дані для створення індивідуального облікового запису;
- б) списки груп користувачів в форматі Excel;
- в) дані для створення груп.

ВИХІДНІ ДАНІ

Вихідна інформація додатку:

- а) списки груп в середовищі кафедри в форматі Excel;
- б) списки облікових записів.

ДОДАТОК Д

Інформаційне середовище кафедри АПЕПС на базі Office 365

Обліковий запис адміністратора

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТР5161_19Б 12-2

Аркушів 2

Київ 2019

ОБЛІКОВИЙ ЗАПИС АДМІНІСТРАТОРА

Цей файл містить дані до облікового запису адміністратора кафедри у середовищі Office 365.

ДОДАТОК Е

Інформаційне середовище кафедри АПЕПС на базі Office 365

Тези на конференцію “Сучасні проблеми наукового забезпечення енергетики”

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТР5161_19Б

Аркушів 3

Київ 2019

На сьогоднішній день однією з найважливіх цілей використання мережі Інтернет є комунікація. Щоб задовольнити цю потребу бізнесу та соціальних інститутів компанії-розробники сучасного програмного забезпечення, наприклад Microsoft, створюють різноманітні середовища для комунікацій. Додатково вони надають можливості використовувати свої API (Application Programming Interface) для налаштування існуючих та створення власних середовищ на базі запропонованих. Ефективна комунікація – це та складова взаємодії, якої досі бракує в сучасних інститутах в межах нашої країни. Саме мається на увазі не просто наявність окремого локального сайту факультету та поштової скриньки (кожен викладач пропонує свій спосіб зв'язку, як результат – відсутня інтеграція в єдину систему), а системне використання на постійній основі повноцінного середовища, яке б дозволяло заводити у систему користувачів різних груп та повноважень (викладачів, студентів, співробітників тощо), організувати діяльність відповідно до типу користувача, вести календарі локальних та загальних подій, викладати та перевіряти завдання індивідуально для кожного студента.

Для вирішення даної задачі пропонується програмний функціонал, який розробляється на базі запропонованого Microsoft хмарного рішення Office 365 [1]. Він надасть можливість викладачам ефективно взаємодіяти зі студентами та іншими викладачами через єдине інформаційне середовище. Також стане можливим динамічно наповнювати це середовище різноманітними матеріалами, новинами, об'явами подій. Викладачі зможуть ефективно працювати не тільки з групами студентів [2], а й окремо з кожним студентом. Це могло б посприяти індивідуальному розвитку студентів, оскільки викладач мав би історію спілкування та відслідковування прогресу в навчанні кожного студента.

Також передбачено розробити зручний інтерфейс органайзера. Його функціонал націлений не тільки на занесення розкладу занять, консультацій та екзаменів, а також для занесення інших видів діяльності (вчена нарада, конференція тощо). Звичайно, можливо додавати інші події, такі як відпустка чи лікарняний. Також можливо організовувати та координувати події, спілкуючись з їх учасниками через вказане середовище.

Дана система може бути широко використана на локальному рівні в різних учбових закладах, тобто не тільки в ВНЗ, де пройде її опробування. Така можливість дозволить вирішити проблеми своєчасного доступу до даних та ефективності комунікації. В першу чергу це б скоротило витрати на пошук необхідної інформації, та відповідним чином збільшило б час на її опанування. Більш того, така система дозволила б ефективніше підійти до навчання студентів за рахунок кращого індивідуального спілкування. Це б дозволило викладачам більш точно оцінювати індивідуальний прогрес студента, що, в свою чергу, призвело б до більш об'єктивної рейтингової системи. Виходячи з наведених прикладів, стає зрозуміло, що таке інформаційне середовище буде використовуватися для вирішення актуальних на сьогоднішній день проблем з розподіленням і використанням часу студентів і викладачів на навчальний процес.

Перелік посилань:

Microsoft SharePoint [Електронний ресурс] – Режим доступу до ресурсу:
<https://products.office.com/uk-ua/sharepoint/collaboration>.

Default SharePoint groups [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.microsoft.com/en-us/sharepoint/default-sharepoint-groups>.